

# **ASPECT: An Advanced Specified-Profile Evaluation Code for Tokamaks**

D. P. Stotler, W. T. Reiersen, and Glenn Bateman  
Princeton Plasma Physics Laboratory  
Princeton University  
Princeton, New Jersey 08543

## **ABSTRACT**

### **Abstract**

A specified-profile, global analysis code has been developed to evaluate the performance of fusion reactor designs. Both steady-state and time-dependent calculations are carried out; the results of the former can be used in defining the parameters of the latter, if desired. In the steady-state analysis, the performance is computed at a density and temperature chosen to be consistent with input limits (e.g., density and beta) of several varieties. The calculation can be made at either the intersection of the two limits or at the point of optimum performance as the density and temperature are varied along the limiting boundaries. Two measures of performance are available for this purpose: the ignition margin or the confinement level required to achieve a prescribed ignition margin. The time-dependent calculation can be configured to yield either the evolution of plasma energy as a function of time or, via an iteration scheme, the amount of auxiliary power required to achieve a desired final plasma energy.

*Classification:* 19.9 Magnetic Confinement

## PROGRAM SUMMARY

*Title of program:* ASPECT

*Catalog number:*

*Program obtainable from:* Daren Stotler, PPPL (Internet: dstotler@theory.pppl.gov)

*Computers:* CRAY-2, CRAY-YMP, Sun SPARC2; *Installations:* National Energy Research Supercomputing Center (NERSC), Lawrence Livermore Laboratory; Theory workstation cluster, PPPL.

*Operating systems:* UNICOS, Sun OS

*Programming language used:* FORTRAN

*Memory required:* 152000 words

*Number of bits in a word:* 64

*Number of processors used:* 1

*Peripheral used:* disk

*Number of lines in distributed program:* 7923

*Keywords:* thermonuclear fusion, tokamak reactor, specified-profile transport

### *Nature of physical problem*

The purpose of this code is to provide a quick, if largely empirical, assessment of the performance of a tokamak fusion reactor [1,2]. ASPECT first performs a steady-state analysis at a point in the density - temperature operating space determined according to a set of user-specified criteria. The code then carries out a time-dependent calculation which can be used to address, for example, auxiliary heating requirements and helium ash buildup.

### *Method of solution*

For the steady-state portion of the calculation, the individual terms in the global power balance equation are estimated using user-specified radial plasma profiles. An expression for the energy confinement time is required; a number of popular

forms are included in the code. The power balance equation is solved using a Brent algorithm subroutine [3]. The reactor performance is quantified using either the ignition margin parameter or the level of confinement required to achieve a specified ignition margin.

For the time-dependent calculation, the temporal variation of the device parameters, the relative plasma density, and the relative auxiliary power must be input. A feedback procedure for limiting the total heating power in the plasma is available. Helium ash accumulation can be included with an adjustable global confinement time. The time-dependent global plasma evolution equations are integrated using Hindmarsh's LSODE algorithm [4]; this yields the plasma energy as a function of time. The integration can be iterated (via the Brent algorithm subroutine [3]) to determine the level of auxiliary input power required to reach a specific plasma energy at a certain time.

#### *Restrictions on the complexity of the problem*

Although ASPECT is designed for use with tokamak reactors, it should be applicable to any toroidal reactor device. It is limited, however, in that it considers only closed flux surfaces; this implies that the cross-section of devices with separatrices are modeled approximately. The behavior of the scrape-off layer plasma is not considered either. A limit of some sort on the total heating power resulting from plasma-wall interactions would be desirable [5]. The code is also restricted in that it cannot treat time-varying profiles.

#### *Typical running time*

1 – 7 CRAY-2 CPU seconds.

#### *Unusual features of the program*

The steady-state calculation provided by ASPECT is considerably more flexible than that provided by similar programs. The ability to solve for the required auxiliary power within the time-dependent calculation is also noteworthy.

## *References*

- [1] W. A. Houlberg, S. E. Attenberger, and L. M. Hively, Nucl. Fusion 22 (1982) 935.
- [2] N. A. Uckan and J. Sheffield, in: Tokamak Startup, ed. H. Knoepfel (Plenum Press, New York, 1986) p. 45.
- [3] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes (Cambridge Press, New York, 1988).
- [4] A. C. Hindmarsh, in: Scientific Computing, ed. R. S. Stepelman (North-Holland, Amsterdam, 1983) p. 55.
- [5] R. J. Goldston et al., Fusion Technol. 21 (1992) 1050.

# LONG WRITE-UP

## 1. Introduction

Over the last several years, simple global performance calculations[1–9] have been used to aid in the design of future tokamak fusion reactors. Given information about the parameters of the device, and some assumptions about the plasma (e.g., plasma shape, radial density and temperature profiles, quality of energy confinement, plasma composition), these codes can determine the fusion yield. These programs are generally small and fast, and are thus suitable for incorporation into larger, more ambitious codes. A primary drawback to the global approach is that there is no general theoretical basis for predicting the plasma parameters or the level of energy confinement. Instead, extrapolations from present empirical data must be made.[2, 10] More detailed scenarios for reactor operation can be obtained by solving the radial transport equations.[11, 12] While the assumptions made in these sorts of calculations are of a more fundamental nature, they as of yet have no firm theoretical foundation either (although some progress has been made recently; see, for example, Ref. 10). Because of the larger size and greater detail of radial transport codes, however, they are generally unsuitable for extensive scans of reactor parameter space, a frequent application of global performance codes.

We will describe here the ASPECT code. ASPECT is an acronym for Advanced Specified-Profile Evaluation Code for Tokamaks. By “specified-profile” we mean that the user determines on input the radial profiles of the plasma density and temperature. The code is intended to “evaluate” tokamak reactor designs, providing some estimate of their performance. We describe the code as “advanced” because of the greater number of options available to the user in comparison with related codes.[1–10] This program was developed so that it could serve as a plasma performance subroutine within a much larger engineering systems code; we discuss here the stand-alone version of the ASPECT code.

This global analysis program consists of two largely distinct parts. In both halves, the user must provide some information about the radial plasma profiles, the plasma composition, and the level of energy confinement. The first half solves a steady-state power balance equation to obtain a unique operating point (i.e., plasma density and temperature). The user also selects a pair of operational limits which delineate limiting boundaries in density and temperature space. Examples of limits incorporated into the code thus far are limits on: density, beta (defined as

the ratio of the plasma to the magnetic pressure), heating power, and temperature. The reactor performance can either be evaluated at the intersection of these limiting boundaries, or the parameter space can be scanned to determine the optimum performance within the limits. The performance of the reactor may be quantified by the ignition margin (at a given confinement level) or by the confinement level required to achieve a specified ignition margin.

The second half of the code solves a time-dependent generalization of this same power balance equation. The results of the steady-state calculation are incorporated into the time-dependent portion by default. However, the input for this half of the program is provided via a separate namelist and may be specified independently. Two modes of operation are available to the user. In the first, the time-dependent equation is integrated a single time, yielding the plasma energy as a function of time during the discharge. In the second, the integration process is iterated upon with various values of the auxiliary heating power, allowing the code to solve for the auxiliary heating power required to achieve a specified value of the plasma energy at the end of the integration. An equation for the helium ash level is integrated simultaneously so that issues such as ash build-up can be addressed; typically, however, a constant ash concentration is assumed, and this feature is disabled.

A feedback loop can be used to modulate the applied auxiliary heating power so that the total heating power remains below some maximum value. The definition of total heating power is chosen on input from three possibilities.

The rest of this document is organized as follows. The global analysis model will be discussed in detail in Sec. 2; various aspects of the model are described in individual subsections. The capabilities and operation of both halves of the code are outlined in Sec. 3. A number of special properties of our global model that allow the steady-state and time-dependent equations to be solved numerically in a convenient manner are presented in Sec. 4. A sample run of the ASPECT code, including input and output, is contained in Sec. 5. The input variables are discussed in the first two appendices. The functions and subroutines which make up ASPECT are briefly described in App. C. Appendix D provides the user with guidance on how to use the information contained in the error number provided by the steady-state part of the code in unsuccessful calculations. Finally, a rough outline of the subroutine calling sequence is presented in App. E.

## 2. Global Analysis Model

Global analysis codes typically solve a steady-state power balance equation similar to

$$P_\alpha + P_{OH} + P_{aux} = P_{con} + P_{rad}. \quad (1)$$

The individual terms represent the volume-integrated contributions made to the total power balance by alpha, ohmic, and auxiliary heating; thermal conduction and radiated losses are on the right-hand side. The steady-state portion of our code will focus on this equation.

Equation (1) is just a special case of a more eneral expression that includes the time-dependence of the plasma energy  $W_{tot}$ . It requires only a little extra effort to consider at the same time the potentially important effect of helium ash accumulation. Hence, we will examine in the time-dependent portion of our code the following coupled set of equations,

$$\frac{dW_{tot}}{dt} = -P_{con} - P_{rad} + P_\alpha + P_{OH} + P_{aux}, \quad (2)$$

$$\frac{dN_{He}}{dt} = C_{ash} \left( \frac{P_\alpha}{k_B E_\alpha} - \frac{N_{He}}{\tau_{p,He}} \right). \quad (3)$$

The various terms in Eqs. (1) and (2) will be described in Sec. 2.3. In Eq. (3),  $N_{He}$  is the total number of helium ash particles in the plasma,  $E_\alpha = 3.5 \times 10^3$  keV is the alpha birth energy,  $k_B = 1.6021 \times 10^{-16}$  J / keV is Boltzmann's constant, and  $\tau_{p,He}$  is the (constant) helium ash particle confinement time. The overall multiplier  $C_{ash}$  is used to enable and disable this feature in the analysis. Equation (3) assumes that the slowing down of fast alpha particles takes place instantaneously. In reality, the alpha slowing-down time in a reactor is expected to be on the order of 0.1 s (depending on plasma parameters).

### 2.1. Shape Dependent Quantities

The shape of the plasma is described by the elongation  $\kappa$  and triangularity  $\delta$ . They are primarily used in computing the plasma volume, cross-sectional area, and safety factor. In particular, the coordinates of the plasma boundary  $R_b$  and  $z_b$  are assumed to be

$$R_b = R + a \cos(\theta + \delta \sin \theta), \quad (4)$$

and

$$z_b = \kappa a \sin \theta, \quad (5)$$

where  $R$  and  $a$  are the plasma major and minor radii, respectively, and  $\theta$  is a poloidal angle parameter (not the true poloidal angle). Note that these expressions cannot describe a separatrix. Consequently,  $\kappa$  and  $\delta$  are usually interpreted as being the elongation and triangularity of the 95% flux surface in a diverted tokamak geometries.

The plasma cross-sectional area  $A_x$  and volume  $V$  are computed numerically from the integrals

$$A_x \equiv 2 \int_0^{\kappa a} dz \int_{-a}^a dx, \quad (6)$$

and

$$V \equiv 2 \int_0^{\kappa a} dz \int_{-a}^a 2\pi R_b dx, \quad (7)$$

where  $x \equiv R_b - R$ .

The cylindrical equivalent safety factor is calculated using the expression[8]

$$q_{cyl} = \frac{\pi a^2 B_T}{\mu_0 I_p R} [1 + \kappa^2 (1 + 2\delta^2 - 1.2\delta^3)], \quad (8)$$

where  $B_T$  is the toroidal magnetic field,  $I_p$  is the plasma current, and  $\mu_0$  is the permeability of free space. An alternative estimate of the cylindrical safety factor, often referred to as the engineering safety factor, is given by

$$q_{eng} = \frac{\pi a^2 \kappa B_T}{\mu_0 I_p R}. \quad (9)$$

Incorporating in an approximate fashion the effects of finite aspect ratio yields an estimate of the MHD safety factor at the 95% flux surface

$$q_{95} \simeq q_{cyl} \frac{1.17 - 0.65a/R}{[1 - (a/R)^2]^2}. \quad (10)$$

In some instances, it is desirable to specify  $q_{95}$  and have the code determine the value of  $I_p$ . In these cases,  $q_{cyl}$  is computed from

$$q_{cyl} = \frac{q_{95}}{C_{I_p}} \frac{[1 - (a/R)^2]^2}{(1.17 - 0.65a/R)}, \quad (11)$$

where  $C_{I_p}$  is an adjustable coefficient ( $= 1$  by default) used to calibrate this expression to the results of a numerical equilibrium. The code then inverts Eq. (8) to obtain  $I_p$ .

## 2.2. Profile Shapes

Codes such as ASPECT are zero-dimensional (0-D) in that no radial transport equation is solved. However, they are more correctly referred to as specified-profile since the plasma profiles are all specified on input. Presently, there are two generic forms for these profiles in our code. The first assumes a parabolic dependence of the density, temperature, and plasma current density on radius,

$$Y = Y_0(1 - r^2/a^2)^{\alpha_Y}, \quad (12)$$

where  $Y$  is replaced by  $n$ ,  $T$ , and  $J$ , respectively;  $r$  is the local minor radius. The profile shapes are then determined by the input values of  $\alpha_n$ ,  $\alpha_T$ , and  $\alpha_J$ .

The other profile form attempts to mock-up the effects of frequent sawtooth oscillations[13] which tend to flatten the profiles inside of the mixing radius  $r_{mix} \sim r_{q=1}$  ( $q$  is the MHD safety factor). Nonzero pedestals at the plasma edge are also allowed with this form. A linear behavior is assumed between the  $q = 1$  surface and the edge:

$$Y = \begin{cases} Y_0 & 0 < r < r_{mix} \\ (Y_0 - Y_{edge})(a - r)/(a - r_{mix}) + Y_{edge} & r_{mix} < r < a, \end{cases} \quad (13)$$

The subscript “0” (“edge”) will be used to designate the value at  $r = 0$  ( $r = a$ ). These trapezoidal profiles are used for density and temperature; the current density is taken to be  $J \propto T^{3/2}$  as one would expect in resistive equilibrium with classical resistivity.

A third option combines the parabolic density variation with the trapezoidal temperature profile. Interpretations of empirical profile data have progressed in such a way as to make this combination particularly easy to calibrate.[10]

A number of integrals over these profiles are computed prior to the start of the main calculations. It is assumed for this purpose that all flux surfaces have the same elongation and are concentric. The resulting “form factors” appear in the expressions for the terms in Eqs. (1) and (2) provided in Sec. 2.3. For parabolic profiles these integrals can, with one exception, be performed analytically. Namely, we define

$$g_{line} = \frac{1}{a} \int_0^a dr \frac{n(r)}{n_0}, \quad (14)$$

$$g_n = \frac{2}{a^2} \int_0^a r dr \frac{n(r)}{n_0}, \quad (15)$$

$$\begin{aligned}
& \rightarrow \frac{1}{1 + \alpha_n}, \\
g_{nT} &= \frac{2}{a^2} \int_0^a r dr \frac{n(r)T(r)}{n_0 T_0},
\end{aligned} \tag{16}$$

$$\begin{aligned}
& \rightarrow \frac{1}{1 + \alpha_n + \alpha_T}, \\
g_{n2Th} &= \frac{2}{a^2} \int_0^a r dr \left[ \frac{n(r)}{n_0} \right]^2 \left[ \frac{T(r)}{T_0} \right]^{\frac{1}{2}},
\end{aligned} \tag{17}$$

$$\begin{aligned}
& \rightarrow 1/(1 + 2\alpha_n + \frac{1}{2}\alpha_T), \\
g_J &= \frac{2}{a^2} \int_0^a r dr \frac{J(r)}{J_0},
\end{aligned} \tag{18}$$

$$\begin{aligned}
& \rightarrow \frac{1}{1 + \alpha_J}, \\
g_{ohmic} &= \frac{2}{a^2} \int_0^a r dr \left[ \frac{J(r)}{J_0} \right]^2 \left[ \frac{T(r)}{T_0} \right]^{-\frac{3}{2}}, \\
& \rightarrow 1/(1 + 2\alpha_J - \frac{3}{2}\alpha_J).
\end{aligned} \tag{19}$$

The arrows indicate the values obtained with parabolic profiles.

Throughout this document, we will make use of the following notation for a volume integral:

$$\langle f \rangle \equiv \frac{2}{a^2} \int_0^a r dr f(r). \tag{20}$$

In particular, when discussing density we will usually refer to the volume-averaged electron density  $\langle n_e \rangle$ . Likewise, we will use the density-weighted, volume-averaged temperature  $\langle T \rangle_{dw} \equiv \langle nT \rangle / \langle n \rangle$  when considering the plasma temperature. Note that throughout these calculations we assume that the electron and ion temperatures are equal,  $T_e = T_i \equiv T$ . In terms of these average quantities and the above form factors, the central values for the density and temperature are given by

$$T_0 = \langle T \rangle_{dw} \frac{g_n}{g_{nT}}, \tag{21}$$

and

$$n_0 = \frac{\langle n \rangle}{g_n}. \tag{22}$$

### 2.3. Power Formulas

In this subsection, we consider in detail each of the terms in Eq. (2). The alpha power is computed using

$$P_\alpha = C_\alpha E_\alpha \frac{2V}{a^2} \int_0^a r dr n_D n_T \overline{\sigma v}_{DT}, \quad (23)$$

where  $C_\alpha$  is a constant multiplier (usually  $C_\alpha = 1$ ). The reactivity,  $\overline{\sigma v}_{DT}$ , is calculated with a formula obtained by Hively [Eq. (5) of Ref. 14] in order to ensure correct results in all temperature regimes. Consequently, this integral must be computed numerically for each value of  $\langle T \rangle_{dw}$ .

The ohmic heating power is (all units are MKS with temperatures in keV; exceptions will be noted)

$$P_{OH} = C_{OH} 1.65 \times 10^{-9} Z_{eff} \Lambda_e \gamma_{NC} T_0^{-3/2} J_0^2 V g_{ohmic}, \quad (24)$$

where  $C_{OH}$  is a constant multiplier ( $C_{OH} = 1$  normally),  $Z_{eff}$  is the effective charge,

$$\Lambda_e = 37.8 - \ln(\langle n_e \rangle^{1/2} / \langle T \rangle_{dw}) \quad (25)$$

and is the Coulomb logarithm,  $\gamma_{NC}$  is the neoclassical resistivity enhancement factor (taken to be a constant, = 2.5). The central current density  $J_0$  is given by

$$J_0 = \frac{I_p}{A_x g_J}. \quad (26)$$

Both bremsstrahlung and synchrotron radiation are included in  $P_{rad} = P_{brem} + P_{sync}$ . We take

$$P_{brem} = C_{brem} 1.68 \times 10^4 (n_{e0}/10^{20})^2 (T_0/10)^{1/2} Z_{eff} V g_{n2Th}, \quad (27)$$

where again  $C_{brem}$  is an arbitrary multiplier, usually set to one. For the synchrotron radiation, we use an expression developed by Trubnikov[15],

$$P_{sync} = C_{sync} 6.214 \times 10^{-17} \langle n_e \rangle \langle T \rangle_{dw} B_T^2 \Phi V. \quad (28)$$

The yield coefficient  $\Phi$  is written in the form

$$\Phi = \frac{G_{sync}}{\sqrt{\Lambda}} \sqrt{(1 - \mathcal{R})},$$

where

$$G_{sync} = 5.198 \times 10^{-3} \langle T \rangle_{dw}^{\frac{3}{2}} \left( 1 + 22.61 \frac{a}{R \langle T \rangle_{dw}^{\frac{1}{2}}} \right)^{\frac{1}{2}},$$

and

$$\sqrt{\Lambda} = 7.78 \times 10^{-9} \left( \frac{\langle n_e \rangle a}{B_T} \right)^{\frac{1}{2}};$$

$\mathcal{R}$  is the wall reflectivity. For most applications,  $P_{sync} \ll P_{brem}$  (provided  $\mathcal{R}$  is not too different from unity, as is expected); hence,  $P_{sync}$  is usually excluded from the calculation by setting  $C_{sync} = 0$ .

Finally, the conducted losses are written as

$$P_{con} = C_{con} \frac{W_{tot}}{\tau_E}, \quad (29)$$

where the total thermal plasma energy is given by

$$W_{tot} = \frac{3}{2} k_B \langle n_e \rangle \langle T \rangle_{dw} \left( 1 + \frac{n_i}{n_e} \right) V. \quad (30)$$

Again, the overall multiplier  $C_{con} = 1$  by default. The next subsection will discuss expressions for  $\tau_E$ .

## 2.4. Energy Confinement Time Formulas

It has been standard practice over the last several years to treat ohmic confinement ( $\tau_{OH}$ ) differently than that found in auxiliary heated experiments ( $\tau_{aux}$ ). A smooth connection has been postulated[16] for the transition between the two regimes

$$\tau_E = \left( \tau_{OH}^{-2} + \tau_{aux}^{-2} \right)^{-\frac{1}{2}}. \quad (31)$$

Since this choice for the exponent “2” has no firm theoretical basis, our code allows a more general expression,

$$\tau_E = \left[ (c_{\tau_1} \tau_{OH})^{x_\tau} + (c_{\tau_2} \tau_{aux})^{x_\tau} \right]^{1/x_\tau}. \quad (32)$$

In the limit of an infinitely sharp transition Eq. (32) becomes

$$\tau_E = \min(c_{\tau_1} \tau_{OH}, c_{\tau_2} \tau_{aux}). \quad (33)$$

This form is also available in the code and is used for  $\tau_{aux}$  expressions which have been fit to data that include ohmic and low (as well as high) auxiliary power discharges.[17, 18] Such expressions should be capable of describing the confinement behavior for all heating regimes. In this case, we take the minimum with  $\tau_{OH}$  solely in order to obtain reasonable behavior in regimes where we would not expect  $\tau_{aux}$  to be accurate [for example, when the net input power  $P_{in} \rightarrow 0$ ;  $P_{in}$  is defined in Eq. (46) below].

The arbitrary multipliers  $c_{\tau_1}$  and  $c_{\tau_2}$  (variable `cctau` in the code) have been inserted in the above equations to allow greater versatility in the usage of existing scaling expressions. For example, it has been found that the confinement times in H-mode operation scale roughly as a constant times the corresponding L-mode value (typically  $c_{\tau_2} \sim 2$ ; see, for example, Ref. 10 and references therein). This allows predictions of H-mode behavior to be made in advance of the development of well-established H-mode scalings. The choice of the  $\tau_E$  scaling expressions to be used in Eq. (32) or (33) is made via the input variables  $\ell_{\tau_1}$  and  $\ell_{\tau_2}$  (`ltau` in the code); the values associated with each scaling are given below.

Of course, the code also permits single scaling expressions to be used by themselves. One must take care when using auxiliary heated scalings in this manner to avoid regions of very low  $\langle T \rangle_{dw}$ ; not doing so could lead to very large  $P_{OH}$ , small  $P_\alpha$ , and consequently unphysical results.

We will now list the available scaling expressions in the present version of the code. The only ohmic scaling is the neo-Alcator formula, obtained with  $\ell_{\tau_i} = 1$ , (see, for example, Ref. 8 and references therein)

$$\tau_{NA} = 7 \times 10^{-22} \bar{n}_e a R^2 q_{cyl}, \quad (34)$$

where  $\bar{n}_e$  is the line-averaged electron density (see Sec. 2.5). While other ohmic scalings may work as well, the ohmically heated behavior of reactors is of relatively little importance in comparison with the auxiliary heated performance. Consequently, there has not been a great deal of effort expended recently to refine and unify  $\tau_{OH}$  scalings. We find it expedient to utilize Eq. (34) in all cases, eliminating this degree of freedom from the problem.

There have been *many* L-mode scalings presented in the literature. The few we include explicitly in the code have enjoyed more widespread use than the others. We first list the power law formulas, starting with the Goldston expression[16] ( $\ell_{\tau_i} = 4$ )

$$\tau_G = 6.432 \times 10^{-8} \left( \frac{\bar{A}_i}{1.5} \right)^{0.5} \kappa^{0.5} I_p (10^2 a)^{-0.37} (10^2 R)^{1.75} P_{in}^{-0.5}. \quad (35)$$

The mass scaling ( $\bar{A}_i$ , average ion mass in AMU) was added some years after the original publication.

A more careful analysis of a larger database led to the Kaye-Goldston scaling[19] ( $\ell_{\tau_i} = 3$ ),

$$\begin{aligned} \tau_{KG} = & 10^{-7.56} \kappa^{0.28} B_T^{-0.09} \left( \frac{I_p}{10^3} \right)^{1.24} \left( \frac{\bar{n}_e}{10^{19}} \right)^{0.26} (10^2 a)^{-0.49} \\ & \times (10^2 R)^{1.65} \left( \frac{\bar{A}_i}{\sqrt{2}} \right)^{0.5} \left( \frac{P_{in}}{10^6} \right)^{-0.58}. \end{aligned} \quad (36)$$

This regression was revised a few years later, resulting in the neo-Kaye expression[18] ( $\ell_{\tau_i} = 2$ )

$$\begin{aligned} \tau_{nK} = & 0.0617 \left( \frac{I_p}{10^6} \right)^{1.12} R^{1.3} a^{-0.04} \kappa^{0.28} B_T^{0.04} \\ & \times \left( \frac{\bar{n}_e}{10^{20}} \right)^{0.14} \bar{A}_i^{0.5} \left( \frac{P_{in}}{10^6} \right)^{-0.59}. \end{aligned} \quad (37)$$

A scaling was developed by the T-10 group[18] ( $\ell_{\tau_i} = 9$ ) is used in our code,

$$\tau_{T-10} = 0.09 a R B_T \bar{A}_i^{0.5} \kappa^{0.5} p_{OH}^{0.08} \left( \frac{P_{in}}{10^6} \right)^{-0.4}, \quad (38)$$

where

$$p_{OH} = Z_{eff}^2 \left( \frac{I_p}{10^6} \right)^4 (a R q_{cyl}^3 \kappa^{1.5})^{-1} \quad (39)$$

is an empirical scaling for the ohmic heating power.

While working with the extensive ITER L-mode database, Kaye developed several new scalings. Two were designated as “complex”, referring to the complicated regression procedure used in the analysis[18]. One included all tokamaks in the database and was correspondingly labeled Kaye-All-Complex ( $\ell_{\tau_i} = 5$ ),

$$\tau_{KAC} = 0.0521 \bar{A}_i^{0.5} \kappa^{0.25} \left( \frac{I_p}{10^6} \right)^{0.85} \left( \frac{\bar{n}_e}{10^{19}} \right)^{0.1} B_T^{0.3} a^{0.3} R^{0.85} \left( \frac{P_{in}}{10^6} \right)^{-0.5}. \quad (40)$$

The other treated only the larger tokamaks in the database and was named Kaye-Big-Complex ( $\ell_{\tau_i} = 6$ ),

$$\tau_{KBC} = 0.082 \bar{A}_i^{0.5} \kappa^{0.25} \left( \frac{I_p}{10^6} \right)^{0.85} \left( \frac{\bar{n}_e}{10^{19}} \right)^{0.1} B_T^{0.3} a^{0.8} R^{0.5} \left( \frac{P_{in}}{10^6} \right)^{-0.5}. \quad (41)$$

With the development of the ITER L-mode database, the choice of regression variables and their units became standard. Consequently, it is convenient to incorporate into our code a general scaling expression ( $\ell_{\tau_i} = 8$ ) based on these standard variables,

$$\tau_{txp} = t_1 \bar{A}_i^{t_2} \kappa^{t_3} \left( \frac{I_p}{10^6} \right)^{t_4} \left( \frac{\bar{n}_e}{10^{19}} \right)^{t_5} B_T^{t_6} a^{t_7} R^{t_8} \left( \frac{P_{in}}{10^6} \right)^{t_9}. \quad (42)$$

The exponents  $t_1 \rightarrow t_9$  are specified on input. With this scheme, we have been able to utilize easily any of a large number of L-mode and H-mode scalings in our calculations.

Two offset-linear scalings are sufficiently popular to warrant inclusion in the code's repertoire. The generic form for an offset-linear scaling is  $\tau_{aux} = \tau_{inc} + W_0/P_{in}$ ; formulas must be provided for both  $\tau_{inc}$ , the incremental confinement time, and  $W_0$  which has the units of energy. The Odajima-Shimomura expression ( $\ell_{\tau_i} = 10$ ) is given by[20]

$$\begin{aligned} \tau_{inc,OS} &= 0.085 \kappa a^2 \bar{A}_i^{0.5} \\ W_{0,OS} &= 0.069 \times 10^6 \frac{I_p}{10^6} \left( \frac{\bar{n}_e}{10^{19}} \right)^{0.6} B_T^{0.2} R^{1.6} a^{0.4} \kappa^{0.2} \bar{A}_i^{0.5} G_{OS}, \end{aligned} \quad (43)$$

where

$$G_{OS} = Z_{eff}^{0.4} \left( \frac{15 - Z_{eff}}{20} \right)^{0.6} \left[ \frac{3q_{cyl}(q_{cyl} + 5)}{(q_{cyl} + 2)(q_{cyl} + 7)} \right]^{0.6}. \quad (44)$$

The other offset-linear expression was developed by Rebut-Lallia[21] ( $\ell_{\tau_i} = 11$ ),

$$\begin{aligned} \tau_{inc,RL} &= 0.017 \frac{I_p}{10^6} L^{1.5} \bar{A}_i^{0.5} Z_{eff}^{-0.5} \\ W_{0,RL} &= 0.206 \times 10^6 \left( \frac{\bar{n}_e}{10^{20}} \right)^{0.75} \left( B_T \bar{A}_i \frac{I_p}{10^6} \right)^{0.5} L^{2.75} Z_{eff}^{0.25}, \end{aligned} \quad (45)$$

where  $L = (Ra^2 \kappa)^{1/3}$  is an average size parameter.

One other option is available to the user:  $\tau_E = \tau_{th} = \text{constant}$  ( $\ell_{\tau_i} = 7$ ). This can be used by itself or with some other scaling [in Eq. (32) or Eq. (33)]. In the latter case, the constant confinement time forms a prescribed lower bound below which  $\tau_E$  cannot fall.

We usually write for the (net) input power used in the confinement scalings the heating power minus the bremsstrahlung radiated power,

$$P_{in} = P_\alpha + P_{OH} + P_{aux} - P_{rad}. \quad (46)$$

We do not include here, nor in Eqs. (1) or (2), the impurity line radiation arising from incompletely stripped impurity ions. In a reactor, this type of radiation is expected to appear only in the cool edge plasma and should not affect the core plasma confinement. Furthermore, since corrections for impurity line radiation are not usually made in performing regression analyses on the energy confinement time, its effects are in a sense built into the empirical scalings for  $\tau_E$ . So, we include in our expression for  $P_{rad}$  only radiation mechanisms which give rise to centrally peaked losses.

By subtracting direct core plasma losses from the heating power when computing  $P_{in}$ ,  $P_{in}$  becomes the net power flowing into the “good confinement” zone between the core and the edge of the plasma. This practice is standard in 0-D calculations[1–3, 8] and has some basis in more detailed simulations[22].

Physically, it is more satisfying for  $\tau_E$  to depend on purely thermodynamic plasma parameters ( $\bar{n}_e$  and  $W_{tot}$ ). This is the sort of form one would expect to derive from a transport theory in which the fluxes are all determined by local plasma parameters (rather than the total power flow; see, for example, Ref. 22). We can use Eq. (1) to convert expressions like the ones presented above into functions of  $\bar{n}_e$  and  $W_{tot}$ . First, consider a single power law scaling

$$\tau_E(P_{in}) = f_\tau P_{in}^{-\gamma}, \quad (47)$$

where  $f_\tau$  contains all of the nonpower dependence (including the multiplier  $c_\tau$ ). Using the definition of  $P_{con}$  and Eq. (46) for  $P_{in}$ , Eq. (1) becomes

$$P_{in} = \frac{W_{tot}}{\tau_E}. \quad (48)$$

Inserting Eq. (47) into Eq. (48) and solving for  $P_{in}$  yields

$$P_{in} = \left( \frac{W_{tot}}{f_\tau} \right)^{1/(1-\gamma)}. \quad (49)$$

Substituting back into Eq. (47), we obtain finally

$$\tau_E(W_{tot}) = \left( f_\tau W_{tot}^{-\gamma} \right)^{1/(1-\gamma)}. \quad (50)$$

An analogous procedure applied to an offset-linear scaling of the form

$$\tau_E(P_{in}) = c_\tau \left( \tau_{inc} + \frac{W_0}{P_{in}} \right) \quad (51)$$

yields

$$\tau_E(W_{tot}) = \frac{c_\tau \tau_{inc} W_{tot}}{W_{tot} - c_\tau W_0}. \quad (52)$$

Since scaling expressions are based upon nearly steady state data[18] for which  $dW_{tot}/dt \ll P_{in}$ , the energy and power forms fit the database equally well. Hence, there are no empirical reasons to choose one over the other. When the plasma is not in steady state, however, the two forms of  $\tau_E$  yield different values of  $dW_{tot}/dt$  [from Eq. (2)]. In particular, the power form of  $\tau_E$  can nearly double the time required to add a specified increment to the plasma energy relative to that needed with the energy form of the confinement time.[1]

## 2.5. Density Relations

All species are taken to have the same radial profile (discussed in Sec. 2.2). Using the definitions  $n_e \equiv \sum_j Z_j n_j$  and

$$Z_{eff} \equiv \frac{\sum_j Z_j^2 n_j}{n_e} \quad (53)$$

(the sums are performed over all ionic species  $j$  with charge  $Z_j$  and density  $n_j$ ) at fixed  $\langle n_e \rangle$ ,  $Z_{eff}$  and  $n_{He}/n_e$ , and assuming one impurity of charge  $Z$ , we obtain the relations

$$\frac{n_D + n_T}{n_e} = \frac{1}{Z - 1} \left[ Z - Z_{eff} - 2(Z - 2) \frac{n_{He}}{n_e} \right], \quad (54)$$

$$\frac{n_Z}{n_e} = \frac{1}{Z^2} \left( Z_{eff} - \frac{n_D + n_T}{n_e} - 4 \frac{n_{He}}{n_e} \right), \quad (55)$$

and

$$\frac{n_i}{n_e} \equiv \frac{n_D + n_T}{n_e} + \frac{n_{He}}{n_e} + \frac{n_Z}{n_e}. \quad (56)$$

A fixed ratio of  $n_D/(n_D + n_T)$  is used to compute the deuterium and tritium densities from  $(n_D + n_T)/n_e$ .

Three different empirically derived expressions for the tokamak density limit have been incorporated into the code. The oldest of these is the Hugill limit[23]

$$\bar{n}_{e,max} = h \frac{B_T}{R_{q_{cyl}}} 10^{20} \text{m}^{-3}. \quad (57)$$

Values of the line-averaged density are converted to volume-averages using the profile form factor  $g_{line}$ ,

$$\bar{n} = \langle n \rangle g_{line} / g_n. \quad (58)$$

Greenwald[24] examined a large amount of data from several different tokamaks and concluded that the expression

$$\bar{n}_{e,max}^{Greenwald} = \frac{I_p}{\pi a^2} 10^{20} \text{ m}^{-3} \quad (59)$$

did the best job of bringing the data into line; Marfes appeared at about 60 – 80% of this value. On the other hand, JET beam-heated discharges were found to be limited by[25]

$$\bar{n}_{e,max}^{JET} = \frac{2B_T}{Rq_{eng}} 10^{20} \text{ m}^{-3}. \quad (60)$$

This limit also worked reasonably well for ohmically heated cases with pellet injection[25].

The most recent evidence (see, for example, Refs. 26–29) indicates that the density limit should actually be a function of input power, as would be appropriate for a density limit caused by excessive impurity radiation. Formulas of this type have not been incorporated into the code as yet.

## 2.6. Plasma $\beta$ Values

The thermal plasma beta is given by the familiar expression

$$\beta_{th} = \frac{2\mu_0}{B_T^2} \langle n_e \rangle k_B \langle T \rangle_{dw} \left( 1 + \frac{n_i}{n_e} \right). \quad (61)$$

To find the contribution due to fast alphas  $\beta_\alpha$ , we make use of the formulas in Refs. 30 and 31. The critical alpha energy at which heating of the background ions equals that of the electrons is given by

$$\begin{aligned} E_{crit} &= 59.2 \langle T \rangle_{dw} \left( \sum_j \frac{Z_j^2 n_j \Lambda_j}{n_e \Lambda_e A_j} \right)^{\frac{2}{3}} \\ &= 59.2 \langle T \rangle_{dw} \left( \left\{ \left[ \frac{n_D}{n_D + n_T} \frac{\Lambda_D}{A_D} + \left( 1 - \frac{n_D}{n_D + n_T} \right) \frac{\Lambda_T}{A_T} \right] \frac{n_D + n_T}{n_e} \right. \right. \\ &\quad \left. \left. + \frac{n_Z}{n_e} \frac{Z^2 \Lambda_Z}{A_Z} \right\} / \Lambda_e \right)^{\frac{2}{3}}. \end{aligned} \quad (62)$$

For simplicity, we approximate the atomic mass of the impurity ion as  $A_Z \simeq 2Z$ ; of course,  $A_D = 2$  and  $A_T = 3$ . The ion Coulomb logarithms are calculated using[30]

$$\Lambda_j = 45.5 + \ln \left[ \left( \frac{\langle T \rangle_{dw}}{\langle n_e \rangle} \right)^{\frac{1}{2}} \frac{A_\alpha A_j}{A_\alpha + A_j} \right], \quad (63)$$

where  $A_\alpha = 4$ ; the electron Coulomb logarithm is given by Eq. (25).

Having found this critical energy, we compute a critical velocity ratio

$$\frac{v_\alpha}{v_{crit}} = \left( \frac{E_\alpha}{E_{crit}} \right)^{\frac{1}{2}}. \quad (64)$$

We also need the slowing-down time

$$\tau_s = 0.371 \left( \frac{\langle T \rangle_{dw}}{10} \right)^{\frac{3}{2}} \left( \frac{10^{20}}{\langle n_e \rangle} \right) \left( \frac{17}{\Lambda_e} \right). \quad (65)$$

The average fast alpha energy (in Joules) is then calculated using

$$\begin{aligned} \overline{E}_\alpha = \frac{k_B E_\alpha}{2} \left( 1 - \left( \frac{v_{crit}}{v_\alpha} \right)^2 \left\{ \frac{1}{3} \ln \left[ \frac{1 + (v_\alpha/v_{crit})^3}{(1 + v_\alpha/v_{crit})^3} \right] \right. \right. \\ \left. \left. + \frac{1}{\sqrt{3}} \left[ 2 \tan^{-1} \left( \frac{2v_\alpha/v_{crit} - 1}{\sqrt{3}} \right) + \frac{\pi}{3} \right] \right\} \right). \end{aligned} \quad (66)$$

The time for the fast alphas to thermalize is

$$\tau_{th} = \frac{\tau_s}{3} \ln \left[ 1 + \left( \frac{v_\alpha}{v_{crit}} \right)^3 \right]. \quad (67)$$

Finally, defining the alpha source rate density  $S_\alpha \equiv P_\alpha/(E_\alpha V)$ , we can write down expressions for the fast alpha density

$$n_\alpha = S_\alpha \tau_{th}, \quad (68)$$

and the fast alpha  $\beta$

$$\beta_\alpha = \frac{2\mu_0}{B_T^2} \frac{2}{3} \overline{E}_\alpha S_\alpha \tau_s. \quad (69)$$

The total plasma beta is then

$$\beta = \beta_{th} + \beta_\alpha. \quad (70)$$

The above expressions are all valid locally. Strictly speaking, we should compute the volume-averaged  $\beta_\alpha$  by integrating Eq. (69) over radius since each of  $S_\alpha$ ,  $\overline{E}_\alpha$ , and  $\tau_s$  depend on  $r$ . Instead, we have for simplicity inserted the volume-averaged quantities everywhere in Eq. (69). A few comparison calculations indicate that the error incurred is negligible.

Limits on beta are calculated using the Troyon scaling[32, 33]

$$\beta_{max} = g_\beta \frac{I_p/10^6}{a B_T} 10^{-2}, \quad (71)$$

where the Troyon coefficient  $g_\beta \sim 2 - 3$  typically.

### 3. Code Capabilities

The ASPECT code is divided into two logically distinct pieces. The first considers Eq. (1) and determines an optimum operating point according to a user-specified criterion. In the second half, the time-dependent behavior is considered. By default, the code incorporates information from the steady-state analysis. But, in general the plasma conditions, and even the machine parameters, can be different in the two calculations.

Summaries of the input variable names, their default values, and their meaning in the calculations are given in Appendices A and B. A documented sample run is presented in Sec. 5. Here, we describe the effects of the input variables in more detail.

#### 3.1. Steady-State Performance

The choice of density limit is determined by `nmurak`. If `nmurak` = 1, the Hugill expression for the maximum density is used [Eq. (57);  $h$  represents the input variable `hugill`]. Likewise, if `nmurak` = 2, the Greenwald [Eq. (59)] limit is utilized; if `nmurak` = 3 the JET [Eq. (60)] expression is employed. If `nmurak` = 4, the maximum volume-averaged electron density is specified on input by the variable `denmax`.

The second variable, `npwr`, provides another limiting surface in  $\langle n_e \rangle - \langle T \rangle_{dw}$  space, giving the code enough information to uniquely determine an operating point (that is a particular set of  $\langle n_e \rangle$  and  $\langle T \rangle_{dw}$ ). If `npwr` = 1,  $P_\alpha$  is required to be

$\leq \text{xpalmx}$ . Similarly, if

$$\text{npwr} = \begin{cases} 3, & P_\alpha + P_{aux} \leq \text{xpalmx}, \\ 4, & P_\alpha + P_{aux} + P_{OH} \leq \text{xpalmx}, \\ 5, & P_\alpha + P_{aux} + P_{OH} - P_{rad} \leq \text{xpalmx}. \end{cases}$$

If  $\text{npwr} = 2$ , it is assumed that  $\langle T \rangle$  is limited to  $\leq \text{tkvmax}$ . If  $\text{npwr}$  takes any other value (e.g., 0), the total plasma  $\beta$  is forced to remain  $\leq \beta_{max}$  [Eq. (71)].

A final possible combination of the two switches involves setting  $\text{nmurak}$  to something other than 1, 2, 3, or 4 (e.g., 0). In this case, the limit specified by  $\text{npwr}$  is used to determine the maximum density and  $\text{tkvmax}$  sets the maximum  $\langle T \rangle_{dw}$ .

The code is capable of either computing the performance at the intersection of the two limits specified by  $\text{nmurak}$  and  $\text{npwr}$  or scanning along the limiting boundaries to find an optimum operating point. To obtain the former, the user would set  $\text{noptm} = 0$ . By using any other value (e.g.,  $\text{noptm} = 1$ ), the optimization is carried out. As will be demonstrated in Sec. 4, reactor performance is usually maximized by operating at the highest density allowed. Hence, by scanning along the limits in  $\langle n_e \rangle - \langle T \rangle_{dw}$  space, one can obtain the optimum performance over the whole operating space. If the chosen confinement scaling is such that this assumption is violated, the performance will always be computed at the intersection of the limits.

The objective of the steady-state calculation is determined by the switch  $\text{ncig}$ . If  $\text{ncig} \neq 0$  (e.g.,  $\text{ncig} = 1$ ), the code solves for the confinement multiplier  $c_\tau$  needed to achieve a specified ignition margin  $\text{xmi}$ . We define the ignition margin by

$$M_I \equiv \frac{P_\alpha}{P_{con} + P_{rad}}. \quad (72)$$

This is related to the more familiar  $Q \equiv P_{fus}/(P_{aux} + P_{OH})$  (the fusion power  $P_{fus} = 5P_\alpha$ ) by

$$Q = \frac{5M_I}{1 - M_I}. \quad (73)$$

The ignition margin is preferred since it is well-behaved in the ignited regime ( $M_I \geq 1$ ).

The confinement multipliers are defined according to Eqs. (32) and (33). If there is only one scaling for  $\tau_E$ , there is no possibility for confusion. If there are two scalings, the multiplier on the second scaling  $c_{\tau_2}$  is solved for. That is, it is assumed that the two components of Eq. (32) or (33) are indeed an ohmic

and an auxiliary heated scaling. The multiplier then represents the confinement enhancement factor required with the specified  $\tau_{aux}$  to reach the input  $M_I$ .

If *no* scalings are specified,  $\tau_E$  is taken to be a constant. Then,  $c_\tau = \tau_E / \tau_{th}$ , where  $\tau_E$  is the confinement time required to achieve  $M_I = \text{xmi}$ , and  $\tau_{th}$  is a user-specified constant confinement time (given by the variable `tauth`).

An upper bound of  $c_\tau = 10$  is used during the solution procedure; this value is chosen on physical rather than numerical grounds (see Sec. 4). Note in particular that during an optimizing search over  $\langle T \rangle_{dw}$ , there may be at low  $\langle T \rangle_{dw}$  no finite value of  $c_\tau$  for which  $M_I = \text{xmi}$  (e.g.,  $P_\alpha - P_{rad} < 0$ ). At such points, we set  $c_\tau = 10$  and proceed with the search. As long as more reasonable values can be obtained at higher  $\langle T \rangle_{dw}$ , this causes no problem.

If `ncig` = 0, the code uses the input specification(s) for  $c_\tau$  and computes  $M_I$  in the manner desired.

In all cases, the value of  $\tau_E$  required to achieve  $M_I = 1$ ,

$$\tau_{ig} = \frac{C_{con} W_{tot}}{\max[(P_\alpha - P_{rad}), 1 \text{ W}]},$$

is also determined. Note that at  $M_I = 1$ ,  $P_{aux} = -P_{OH}$ , not  $P_{aux} = 0$  as is usually assumed at ignition. This is a trivial point, but is of some importance in the time-dependent calculations.

### 3.2. Time-Dependent Performance

In the simplest situation (`nfindp` = 0), this part of the code integrates Eq. (2) as a function of time using initial conditions and time-dependent parameters specified on input. The result of the calculation is the plasma energy (in effect, the temperature since the density evolution is prescribed) as a function of time  $t$  between initial and final times  $t_{init}$  and  $t_f$ , respectively. The input parameters are chosen via a second namelist that is independent of the first. In fact, the time-dependent portion can analyze a device that is completely different from that considered in the steady-state part of the code.

If `nfindp`  $\neq$  0 (e.g., = 1), the code iterates on  $P_{aux}$  to determine the value required to achieve a desired final energy  $W_{tot,f}$  at time  $t_f$ . Since  $P_{aux}$  is in general time-dependent, the code actually determines its maximum value  $P_{aux,max}$ ; the input array representing the auxiliary power is renormalized after input so that it can be used as  $[P_{aux} / P_{aux,max}](t)$  (the same prescription holds for `nfindp` = 0 except that the user specifies  $P_{aux,max}$ ). By default,  $W_{tot,f}$  is taken to be the total

thermal energy at the operating point calculated by the steady-state portion of the code. In this way, the steady-state calculation can determine an optimum density and temperature, and the time-dependent analysis can compute the auxiliary power required to get there in a specified amount of time. Note that output is produced only for the final value of  $P_{aux,max}$ .

Helium ash accumulation is controlled by the variable `cash` [ $C_{ash}$  in Eq. (3)]. If  $C_{ash} = 0$ , the time-evolution of the helium ash concentration  $n_{He}/n_e(t)$  is specified by the input array `arnane(i)`. If  $C_{ash} \neq 0$ , an initial number of helium ions  $N_{He,0}$  (`xnhein`) is assumed to be present at the start of the calculations, and Eq. (3) is integrated to determine the time evolution of  $N_{He}$ . Note that if  $C_{ash} \neq 0$ , solving for  $P_{aux,max}$  (`nfindp`  $\neq 0$ ) may give unreliable results and is not recommended.

The level of auxiliary power can be modulated via a feedback loop to limit the total input power, as was done in the steady-state part of the code with `npwr`. In this case, setting `nfdbak` = 1, 2, or 3 forces  $P_{aux} + P_\alpha$ ,  $P_{aux} + P_\alpha + P_{OH}$ , or  $P_{aux} + P_\alpha + P_{OH} - P_{rad}$ , respectively, to be  $\leq P_{in,max}$  (`xpinmx` on input). When  $3 \leq npwr \leq 5$ , `nfdbak` is defaulted to `npwr-2`, and  $P_{in,max}$  is set to `xpalmx`+0.3 MW; the additional 0.3 MW provides a little “headroom” to allow  $P_{aux,max}$  to be determined more easily when `nfindp`  $\neq 0$ . If `nfdbak` is something other than 1, 2, or 3, no feedback is performed. Time-delay problems make it difficult to maintain  $P_\alpha = \text{constant}$  via feedback;[34] hence, that option is not available.

The input time-dependent waveforms consist of several time breakpoint arrays and one or more parameter value arrays for each. For example, the array `tbk(i)` provides the time breakpoints for  $a(t)$ ,  $R(t)$ ,  $\kappa(t)$ , and  $\delta(t)$ ; the latter are input via `armin(i)`, `armaj(i)`, `aalong(i)`, and `atring(i)`, respectively. The number of breakpoints is given by `ntbk`, which must be  $\leq 30$  given the present array dimensions. In some instances, it may be desirable to specify a particular  $q_{cyl}(t)$  evolution.[35] If so,  $a(t)$  and  $R(t)$  are calculated from  $q_{cyl}(t)$  and Eq. (8) [input via `aqcyl(i)`] by assuming  $a(t) + R(t) = a_{ss} + R_{ss}$ , where the subscript “ss” denotes the value used in the steady-state calculation.

Density-related parameters are controlled by `tnbk(i)`,  $1 \leq i \leq \text{ntnbk}$ ; the associated parameters are  $\langle n_e \rangle / \langle n_e \rangle_{ft}(t)$  [`adenav(i)`],  $Z_{eff}(t)$  [`azeff(i)`], and  $n_{He}/n_e(t)$  [`arnane(i)`]. Note that determining  $\langle n_e \rangle(t)$  requires two pieces of information. Following the `namelist read` statement,  $\langle n_e \rangle / \langle n_e \rangle_{ft}(t)$  is renormalized to a maximum value of 1. The maximum or “flattop” density  $\langle n_e \rangle_{ft}$  is specified by the variable `denflt`. Thus, the density computed in the steady-state part of the code can be incorporated into the time-dependent calculation via the

default setting  $\langle n_e \rangle_{ft} = \langle n_e \rangle_{ss}$  while still allowing an arbitrary time-dependence.

Time breakpoints for the plasma current  $I_p(t)$  [acurnt(i)] and toroidal field  $B_T(t)$  [abtor(i)] are given by the array t**ibk**(i),  $1 \leq i \leq \text{ntibk}$ . Time-dependent confinement multipliers  $c_{\tau_j}(t)$  ( $j = 1, 2$ ) can be specified via actaue(i, j) according to the time breakpoints set in t**tbk**(i),  $1 \leq i \leq \text{nttbk}$ .

The value of  $P_{aux}(t)$  is specified in much the same way as  $\langle n_e \rangle(t)$ . The overall time-dependence is prescribed by apaux(i) with time breakpoints t**pbk**(i),  $1 \leq i \leq \text{ntpbk}$ ; this array is renormalized to have a maximum value of  $P_{aux,max}$  (xpaumx). If nfindp  $\neq 0$ ,  $P_{aux,max}$  is varied so that the plasma energy achieves  $W_{tot,f}$  at  $t_f$ . Also, if feedback is invoked with nfdbak = 1, for example, and  $P_\alpha(t) + P_{aux}(t) > P_{in,max}$ ,  $P_{aux}(t)$  is reset to  $P_{in,max} - P_\alpha$ .

The routine LSODE[36] is used to integrate Eqs. (2) and (3); the Adams predictor-corrector scheme (chord method with numerically calculated Jacobian) is employed. Actually, three equations are integrated. The third computes the diagnostic  $W_{heat}$  defined as[10]

$$W_{heat}(t) \equiv \int_{t_{init}}^t dt (P_\alpha + P_{aux} + P_{OH}). \quad (74)$$

This represents the integrated power input to the plasma and is thus related to the total heat deposited onto the divertor or limiter targets. In order to obtain a conservative estimate of the actual heat transmitted to the target, radiation is *not* subtracted here.

As LSODE proceeds, it calls a subroutine that evaluates the right-hand sides of the equations for  $dW_{tot}/dt$ ,  $dN_{He}/dt$ , and  $dW_{heat}/dt$  at a particular time  $t$ . A linear interpolation scheme is applied to obtain parameter values when  $t$  falls between the input breakpoints. If the number of breakpoints is set  $\leq 1$  or if  $t <$  the first breakpoint time, the first value in each associated parameter array is used. Likewise, if  $t$  extends beyond the last time breakpoint, the parameter value at that last breakpoint is employed.

The present defaults are similar to those described in the BPX Physics Design Description[10]. The plasma current is ramped up linearly over tramp = 7.5 s; the beginning of the current ramp is designated as  $t = 0$ . The toroidal field is ramped up much more slowly; a bilinear waveform is assumed. The actual values are given in App. B. At the end of the flattop ( $t = \text{tramp} + \text{tburn} = 17.5$  s),  $I_p(t)$  and  $B_T(t)$  fall linearly to 0 and  $\text{zztf3} \cdot B_{T,ss}$ , respectively, at  $t = \text{tramp} + \text{tburn} + \text{tdump} = 25$  s (zztf3 is defined in App. B).

The minor radius is taken to have its full value for  $t \geq 1.5$  s up through the end of the flattop. The evolution of  $a(t)$  at earlier times is generally complicated. Hence, most of our simulations begin at  $t = 1.5$  s. For completeness, a simple linear ramp from  $a(t = 0) = 0$  is prescribed. The default plasma elongation is brought up from 70% of its flattop to value to 100% between  $t = 1.5$  s and  $t = \text{tramp}$ . Similarly, the default triangularity increases from 20% to 100% of its flattop value during the same period. We arbitrarily set the values of these two parameters at  $t = 0$  to be  $\kappa(t = 0) = 1$  and  $\delta(t = 0) = 0$ . During the shutdown,  $a(t)$  and  $\delta(t)$  fall linearly to 0; over this same period,  $\kappa(t)$  drops to 1. The default major radius is chosen so that with the above-prescribed  $a(t)$ ,  $a(t) + R(t) = a_{ss} + R_{ss}$ . This is required for the outer edge of the plasma to remain in contact with the ICRH antenna.

By default, the plasma density rises linearly from 0 during the current ramp-up, and drops back down to 0 during the shutdown. Each of  $Z_{eff}(t) = Z_{eff,ss}$ ,  $[n_{He}/n_e](t) = [n_{He}/n_e]_{ss}$ ,  $\tau_{th}(t) = \tau_{th,ss}$ , and  $c_{\tau_i}(t) = c_{\tau_i,ss}$  are by default constant and take on the values assigned in the steady-state calculation.

The input array describing  $P_{aux}/P_{aux,max}(t)$  is defaulted to 0. Nonetheless, the actual default prescription for the time-dependence of the auxiliary power is nontrivial. Unless some element of the input array  $\text{apaux}(i) > 0$ , the code uses the toroidal field evolution to estimate the radius of the ICRH resonance layer,

$$r_{res} = R(t) \left[ 1 - \frac{B_T(t)}{B_{T,max}} \right]. \quad (75)$$

We assume that the radial variation of the magnetic field is given by  $B_T(r) \propto 1/[R(t) - r]$  (the resonance point moves out in major radius as the toroidal field increases), and that the resonance layer is at the magnetic axis when  $B_T$  achieves its maximum value  $B_{T,max}$ . If  $r_{res} > a(t)/2$ ,  $P_{aux}$  is set = 0. For resonance layers inside this radius, the heating efficiency is taken to be equal to the relative density at the layer:[37]

$$\frac{P_{aux}}{P_{aux,max}} \sim \frac{n_e(r_{res})}{n_e(0)}. \quad (76)$$

Thus, during the flattop, heating will be 100% efficient.

## 4. Basic Properties of the Model

There are several analytically derived properties of this model that allow the solutions to Eq. (2) to be obtained numerically in a convenient fashion.

First consider a situation in which we are solving for the value of  $c_\tau$  required to achieve a certain  $M_I$  (nc ig  $\neq 0$ ). From Eq. (1) and the definition of the ignition margin, Eq. (72),

$$P_{aux} = (P_\alpha + P_{OH}) \frac{1 - M_I}{M_I}. \quad (77)$$

This expression enables us to solve Eq. (1) directly for the value of  $c_\tau$  (which enters into  $P_{con}$  through the definition of  $\tau_E$ ) by eliminating the unknown  $P_{aux}$ .

In cases where the power form of  $\tau_{aux}$  is being used, we can go one step further and obtain a reasonable lower bound on the solution  $c_\tau$ . Namely, if we know  $P_{aux}$ , we can compute  $P_{in}$  [Eq. (46)] for a given  $\langle n_e \rangle$  and  $\langle T \rangle_{dw}$ . The code can then evaluate the auxiliary heated confinement time  $\tau_{aux}$ . By defining  $c'_\tau \equiv \tau_E / \tau_{aux}$ , the code can easily calculate that in steady-state

$$c'_\tau = \frac{W_{tot}}{P_{in} \tau_{aux}}. \quad (78)$$

From Eq. (32),  $\tau_E \leq c_\tau \tau_{aux}$ , thus

$$c'_\tau \leq c_\tau \quad (79)$$

is a lower bound for the desired solution  $c_\tau$ . The Brent algorithm[38] used to solve Eq. (1) also requires an upper bound be specified for the solution. Since we do *not* expect confinement times  $\tau_E \gg \tau_{aux}$  (e.g., H-mode confinement is presently about twice as good as L-mode), we set an upper bound  $c_\tau \leq 10$ .

In cases where the code is solving for  $P_{aux}$  at a given  $c_\tau$ , one bound on the solution can be obtained using the quantities  $P_{in,0}$  and  $P_{aux,0}$  defined as

$$P_{in,0} \equiv P_\alpha + P_{OH} - P_{rad} \quad (80)$$

and

$$P_{aux,0} \equiv \frac{W_{tot}}{\tau_E(P_{in,0})} - P_{in,0}. \quad (81)$$

We can write down an similar expression for  $P_{aux}$  using Eqs. (1) and (46),

$$P_{aux} = \frac{W_{tot}}{\tau_E(P_{in,0} + P_{aux})} - P_{in,0}. \quad (82)$$

Comparing Eqs. (81) and (82), and assuming increased input power degrades confinement, we have

$$P_{aux} \left\{ \begin{array}{l} > \\ < \end{array} \right\} P_{aux,0} \text{ for } P_{aux} \left\{ \begin{array}{l} > \\ < \end{array} \right\} 0. \quad (83)$$

These expressions provide an upper or lower bound for  $P_{aux}$ , depending on its sign. A subtlety of this procedure is that the decision of which case applies must be based on  $P_{aux,0}$  since the sign of  $P_{aux}$  is not known a priori. But, it can be shown that as long as  $d[P_{in}\tau_E]/dP_{in} > 0$  for all  $P_{in}$ ,  $P_{aux} \cdot P_{aux,0} > 0$ .

The optimization procedures used in the steady-state part of the code assume that reactor performance improves monotonically with increasing electron density. We will now outline the conditions under which this assumption is valid.

Defining

$$\gamma \equiv -\frac{\partial \ln \tau_{aux}}{\partial \ln P_{in}}, \quad (84)$$

and

$$\alpha \equiv \frac{\partial \ln \tau_{aux}}{\partial \ln \langle n_e \rangle}, \quad (85)$$

it can be shown that

$$\left. \frac{\partial M_I}{\partial \langle n_e \rangle} \right|_{c_\tau} = \frac{M_I}{\langle n_e \rangle} \frac{P_{con}}{P_{con} + P_{rad}} \left[ \frac{1 - 2\gamma + \alpha + 2(\tau_{aux}/\tau_{NA})^{x_\tau}}{1 - \gamma + (\tau_{aux}/\tau_{NA})^{x_\tau}} \right]. \quad (86)$$

Furthermore,

$$\left. \frac{\partial c_\tau}{\partial \langle n_e \rangle} \right|_{M_I} = -\frac{c_\tau}{\langle n_e \rangle} [1 - 2\gamma + \alpha + 2(\tau_{aux}/\tau_{NA})^{x_\tau}]. \quad (87)$$

Of course, for a simple power law scaling,  $\gamma$  and  $\alpha$  are just the exponents on  $P_{in}$  and  $\langle n_e \rangle$ . Thus, we can see that all of the power law scalings listed in Sec. 2.4 (except neo-Kaye) yield

$$\left. \frac{\partial M_I}{\partial \langle n_e \rangle} \right|_{c_\tau} > 0 \quad (88)$$

and

$$\left. \frac{\partial c_\tau}{\partial \langle n_e \rangle} \right|_{M_I} < 0. \quad (89)$$

The same is typically true for the offset linear scalings discussed in Sec. 2.4. Specifically, the requirement is that  $\tau_E < 3.5\tau_{inc}$  for the Shimomura-Odajima scaling and  $\tau_E < 5\tau_{inc}$  for Rebut-Lallia.

Thus, regardless of whether the code is solving for the maximum  $M_I$  at a given  $c_\tau$  or the minimum  $c_\tau$  at a given  $M_I$ , the optimum density at any value of  $\langle T \rangle_{dw}$  is the largest density allowed by the specified limits. The code then need only

scan  $\langle T \rangle_{dw}$  along this limiting boundary to determine the optimum performance. Optimization is not allowed when neo-Kaye scaling or an externally specified scaling with  $1 - 2\gamma + \alpha < 0$  (i.e.,  $0.5 + \text{txp}(9) + 0.5\text{txp}(5) < 0$ ) is requested. It can also be shown that the extremum of  $M_I$  on the limiting boundary occurs at the same  $\langle T \rangle_{dw}$  as does that of  $c_\tau$ . Namely,

$$\left. \frac{\partial M_I}{\partial \langle T \rangle_{dw}} \right|_{c_\tau} = 0 \Rightarrow \left. \frac{\partial c_\tau}{\partial \langle T \rangle_{dw}} \right|_{M_I}. \quad (90)$$

## 5. Sample Run

In this section, we discuss the input and output from a typical run of the ASPECT code. This case represents a reference operating point for BPX[10].

The format of the file shown in Fig. 1 is appropriate for use with namelist input under the UNIX operating system. All characters on a line after a “!” are ignored; this facility makes it possible to document the input file as shown below. The name of the input file must be `plsinp`; both namelists are read from this file.

There are a few items of particular interest in this input file.

- Setting `cip = 0.` allows the value of  $I_p$  to be read in through `curma`.
- The subroutine `STRIPX` will remove everything in a line after a “!”. This allows the input file to be commented in the manner shown.
- The 1989 ITER-P L-mode scaling[39] is used here. Note that the comments are particularly convenient in helping the reader remember which exponents go with which variables.
- The namelists are terminated by `$end`.
- In the second namelist, all time-dependent variables have their default values except for  $\langle n_e \rangle / \langle n_e \rangle_{max}$  and  $P_{aux} / P_{aux,max}$ . The flattop density is taken from the result of the steady-state calculation.
- We set `dtout = 0.5 s` to make insertion of the output in this document easier. If the time-dependent output is to be plotted, a smaller value (e.g., `dtout ≤ 0.2 s`) would be more desirable.

The steady-state calculation write its output to a file called `pl sout` following an “echo” of the input file. The results obtained from the sample run are presented in Fig. 2.

We should point out that

- The current  $I_p$  is shown since it is calculated if `cip`  $\neq$  0.
- The second line provides  $\langle n_e \rangle$  (in  $\text{m}^{-3}$ ) and  $\langle T \rangle_{dw}$  (in keV).
- The third and fourth lines contains the power levels in Eq. (1),  $P_\alpha$ ,  $P_{brem}$ ,  $P_{con}$ ,  $P_{sync}$ ,  $P_{aux}$ , and  $P_{OH}$ , in that order. All are in watts.
- The fifth line gives the total beta, the beta due to fast alpha particles, and the energy confinement time  $\tau_E$  (in seconds).
- In the sixth line the computed values for  $c_\tau$  (an input in this case),  $M_I$ , and  $\tau_{ig}$  (in seconds) are presented.
- Finally, the last line shows the error parameter; it equals zero for a succesful calculation. The meaning of nonzero values is discussed in Sec. D.

The time-dependent calculation produces columnar output in the file `tdout`; the output obtained with the sample input file is presented in Fig. 3. From left to right the columns represent: the time during the discharge  $t$  (in seconds),  $\langle n_e \rangle$  (in units of  $10^{20} \text{ m}^{-3}$ ),  $\langle T \rangle_{dw}$  (in keV),  $P_\alpha$ ,  $P_{aux}$ ,  $P_{OH}$ ,  $P_{con}$ ,  $P_{rad} = P_{brem} + P_{sync}$  (all in mega-watts),  $W_{tot}$  (in mega-joules),  $(n_D + n_T)/n_e$ , and  $W_{heat}$  (in mega-joules). No graphics are produced directly by the ASPECT code; we instead format this output so that it can be easily “pasted” into a graphing or spreadsheet application running on an Apple Macintosh computer. A plot produced using this data is shown in Fig. 4.

The value of  $P_{aux,max}$  and the value of  $W_{tot}(t_f)$  at are listed separately at the end of the file for clarity.

## Acknowledgments

This work was supported by U.S. DOE Contract No. DE-AC02-76-CHO-3073.

## Appendix A. First Namelist Input Variables

Variables input via the first namelist control the steady-state calculation. They are described in the following table. The first column gives the variable name. Its default value and units are given in the second column. If assigned a symbol in the text, it is indicated in the third column; the equation or section in which it is defined (if applicable) is shown in the fourth column. The last column provides a brief description.

**You can find this table and the one associated with the next appendix at the end of the document.**

## Appendix B. Second Namelist Input Variables

The variables in the second input namelist control the time-dependent calculation. They are described in the following table. The first column gives the variable name. Its default value and units are given in the second column. If assigned a symbol in the text, it is indicated in the third column; the equation or section in which it is defined (if applicable) is shown in the fourth column. The last column provides a brief description. Some of the defaults for this namelist are specified by variables. Some of these will correspond to input parameters from the first namelist. Others will refer to values determined during the steady-state calculation; these are denoted by a subscript “ss”. Second namelist variables appearing in the default specification of another variable will take on their default values during the calculation. A handful of externally defined variables appear in the default specifications of the field ramps for historical reasons. Their names and corresponding values are:  $zztf2 = 0.748$ ,  $zztf3 = 0.593$ ,  $zztf4 = 0.889$ ,  $tramp = 7.5$  s,  $tdump = 7.5$  s,  $t_{null} = tramp/1.9445$ , and  $t_{burn} = 10$  s.

## Appendix C. Functions and Subroutines

The following is a list of the subroutines included in the ASPECT code and a brief description of their purpose. Note that the routines beginning with PL are part of the steady-state calculation, while those starting with TD form the time-dependent part. More details can be found in the internal documentation of the code.

- RUNPLS Sets defaults, reads namelist, and initiates calculations.
- PLMAIN Controls steady-state calculations.
- PLHERE Evaluates performance at maximum  $\langle n_e \rangle$  and  $\langle T \rangle_{dw}$  allowed by limits.
- PLOPTM Searches  $\langle T \rangle_{dw}$  to determine optimum performance allowed by limits.
- PLMICT Given  $\langle T \rangle_{dw}$ , evaluates limiting  $\langle n_e \rangle$  and computes performance.
- PLFMI Calculates the ignition margin  $M_I$ .
- PLCTAU Calculates the confinement time multiplier  $c_\tau$  required to achieve the specified  $M_I$ .
- PLFNDN Finds maximum  $\langle n_e \rangle$  within the prescribed limits.
- PLBETC Computes  $\beta_{th} + \beta_\alpha - \beta_{max}$ .
- PLPHTC Evaluates the difference between the actual and limiting heating powers.
- PLPRT2 Auxiliary routine for PLPHTC to differentiate forms for the heating power.
- PLPOWC Determines difference between power inputs and losses.
- TDMAIN Controls time-dependent calculations.
- TDELW Integrates differential equation to find  $W_{tot}(t_f)$ .
- TDWDOT Evaluates  $dW_{tot}/dt$ ,  $dN_{He}/dt$ , and  $dW_{heat}/dt$ .
- TDWJ Dummy function for LSODE.

TDGET    Determines plasma parameters at a given time.

TDPAUX   Evaluates the auxiliary power at a given time.

TDINTP   Linearly interpolates input waveforms between time breakpoints.

SLSODE   Package containing Hindmarsh’s LSODE subroutine[36].

In addition a library of utility routines is used. This library, called POPHYS, includes:

BRENT    Determine the minimum of a function using Brent’s method. [38]

EVBeta   Evaluates  $\beta$  quantities and fast alpha to electron density ratio.

EVTAE    Evaluates the energy confinement time.

PALPHA   Computes  $P_\alpha$  based on Hively’s fit.[14]

SETCON   Sets commonly used constants.

SETFRM   Computes profile form factors.

SETPOW   Calculates nonconduction terms in the power balance.

SIMPSN   Calculates a definite integral using Simpson’s rule.

STRIPX   Strips comments following “!” from input files.

VOLARE   Computes volume, cross-sectional area, surface area, and arc length.

ZBRENT   Finds root of a function using Brent’s method.[38]

## Appendix D. Error Codes

The error code reported in the output following a failed steady-state calculation contains information about the origin of the error condition. In fact, the code is a string of digits, one for each routine in the subroutine calling chain (see App. E). The rightmost digit corresponds to the routine highest in the chain (PLMAIN). The leftmost digit is generated by the routine responsible for initiating the error condition. The value of each digit indicates which error checking

statement in that routine detected the error. In the code, these can be identified by searching for assignments of the variable `kerr`. Thus, one can track down an error by back-tracking along the calling chain, using the error code to locate the problem in each subroutine.

In the time-dependent calculation, specific error messages are provided in the output.

## Appendix E. Calling Sequence

The following outline provides a general description of the logic of the code and the order in which the routines are called. For simplicity, many details are left out here. But, enough information is provided to allow the reader to answer any further questions by referring to the code itself. Subroutines not listed here are elementary enough in their function that no detail beyond the descriptions given in App. C is necessary.

### Main Program

- A. Call STRIPX
- B. Call RUNPLS:
  - 1. Set defaults.
  - 2. Read first namelist.
  - 3. Call VOLARE
  - 4. Call PLMAIN
  - 5. Write steady-state output.
  - 6. Call TDMAIN
  - 7. Return
- C. Stop

### PLMAIN

- A. Call SETCON
- B. Call SETFRM
- C. Calculate  $I_p, q_{cyl}$ .
- D. Evaluate  $\beta_{max}$  [Eq. (71)],  $\langle n_e \rangle_{max}$  [Eqs. (57), (59), and (60)].

E. Call PLHERE (noptm = 0) or PLOPTM (otherwise)

F. Return

#### I. PLHERE

A. If  $1 \leq \text{nmurak} \leq 4$ , set  $\langle n_e \rangle = \langle n_e \rangle_{max}$ , and

1. If npwr = 1 or  $3 \leq \text{npwr} \leq 5$ , use ZBRENT to solve PLPHTC = 0. for  $\langle T \rangle_{dw}$ ,
2. Else if npwr = 2, set  $\langle T \rangle_{dw} = \text{tkvmax}(\text{input})$ ,
3. Else, use ZBRENT to solve PLBETC = 0. for  $\langle T \rangle_{dw}$ .

B. Else, set  $\langle T \rangle_{dw} = \text{tkvmax}(\text{input})$ , and

1. Use PLFNDN to solve for  $\langle n_e \rangle$ .

C. Call PLMICT

D. Return

#### II. PLOPTM

A. Bracket a minimum of PLMICT between  $\langle T \rangle_{dw} = 1 \text{ keV}$  and tkvmax.

B. Use BRENT to solve for minimum in PLMICT  $\Rightarrow \langle T \rangle_{dw}$ .

C. Return

#### III. PLFNDN

A. Set  $\langle n_e \rangle = \langle n_e \rangle_{max}$ ;

1. If noptm  $\neq 0$ , and  $\langle n_e \rangle$  is *below* specified power or  $\beta$  limit, Return.

B. If npwr = 1 or  $3 \leq \text{npwr} \leq 5$ , solve PLPHTC = 0. for  $\langle n_e \rangle$ ,

C. Else, use ZBRENT to solve PLBETC = 0. for  $\langle n_e \rangle$ .

D. Return

#### IV. PLMICT

A. If noptm  $\neq 0$ , then

1. If  $1 \leq \text{nmurak} \leq 4$ , and npwr = 2, set  $\langle n_e \rangle = \langle n_e \rangle_{max}$ .

2. Else, use PLFNDN to solve for  $\langle n_\epsilon \rangle$ .

B. Call SETPOW

C. If  $\text{ncig} \neq 0$ ,  $c_\tau = \text{PLCTAU}(M_I)$ ,

D. Else,  $M_I = -\text{PLFMI}(c_\tau)$ .

E. Return

#### V. PLCTAU

A. Evaluate  $P_{aux}(M_I)$  [Eq. (77)].

B. If  $\text{npwt}c \neq 0$ , evaluate lower bound for  $c_\tau$  [Eq. (78)],

C. Else, set lower bound to  $c_\tau = 0.1$ .

D. Use ZBRENT to solve  $\text{PLPOWC} = 0$ .

E. Return

#### VI. PLFMI

A. If  $3 \leq \text{npwr} \leq 5$ , may have determined  $P_{aux}$  in solving  $\text{PLPHTC} = 0$ .,

1. Evaluate  $\text{PLPOWC}$ .

2. If  $\text{PLPOWC} < 10^3 \text{ W}$ , Return.

B. Evaluate  $P_{aux,0}$  [Eq. (81)].

C. Use as upper or lower bound, depending on sign of  $P_{aux,0}$ .

D. Use ZBRENT to solve  $\text{PLPOWC} = 0$ . for  $P_{aux}$ .

E. Calculate  $M_I$  [Eq. (72)].

F. Set  $\text{PLFMI} = -M_I$  (use “-” since BRENT solves for minimum).

G. Return

#### VII. PLBETC

A. Evaluate  $\beta$  [Eq. (70)].

B. Set  $\text{PLBETC} = \beta - \beta_{max}$ .

C. Return

#### VIII. PLPHTC

A. Call SETPOW

- B. If  $\text{npwr} = 1$ , set  $\text{PLPHTC} = P_\alpha - \text{xpalmx}$ ,
- C. Else,
  - 1. Call PLPRT2:
    - a. If  $\text{npwr} = 3$ ,  $P_2 = P_{OH} + \text{xpalmx}$ ,
    - b. If  $\text{npwr} = 4$ ,  $P_2 = \text{xpalmx}$ ,
    - c. Else ( $\text{npwr} = 5$ ),  $P_2 = \text{xpalmx} + P_{rad}$ .
  - 2. If  $\text{ncig} = 1$ ,
    - a. Set  $P_1 = P_\alpha / M_I$ .
    - b. Set  $\text{PLPHTC} = P_1 - P_2$ .
  - 3. Else,
    - a. Set  $P_1 = P_\alpha + P_{OH}$ .
    - b. Set  $P_{aux} = P_2 - P_1$ .
    - c. Set  $\text{PLPHTC} = \text{PLPOWC}(P_{aux})$ .
  - 4. Return

#### IX. PLPOWC

- A. Calculate  $P_{in}$  [Eq. (46)].
- B. Set  $\tau_E = \text{EVTAUE}(P_{in})$ .
- C. Evaluate  $P_{con}$  [Eq. (29)].
- D. Set  $\text{PLPOWC} = P_{con} + P_{rad} - (P_\alpha + P_{OH} + P_{aux})$ .
- E. Return

#### TDMAIN

- A. Set defaults.
- B. Read second namelist.
- C. Rescale density waveform.
- D. If  $\text{nfindp} = 0$ , evaluate  $W_{tot}(t_f)$  using  $\text{TDELW}(P_{aux,max})$  (set  $W_{tot,f} = 0$ .),
- E. Else, use ZBRENT to solve  $\text{TDELW} = 0$ .
- F. Return

#### I. TDELW

- A. Calculate  $W_{tot}$  and  $N_{He}$  at  $t_{init}$  from input.
- B. Loop over output time-steps  $\Delta t$ :
  - 1. Call LSODE to integrate TDWDOT over next  $\Delta t$ .
  - 2. Write output if this is the last call to TDELW.
- C. Set  $TDELW = W_{tot}(t_f) - W_{tot,f}$ .
- D. Return

## II. TDWDOT

- A. Call TDGET
- B. Call SETPOW
- C. Use TDP AUX to evaluate  $P_{aux}$ .
- D. Limit  $P_{aux}$  by feedback if needed.
- E. Calculate  $P_{in}$  [Eq. (46)].
- F. Set  $\tau_E = EVTAUE(P_{in})$ .
- G. Evaluate  $P_{con}$  [Eq. (29)].
- H. Calculate  $dW_{tot}/dt$ ,  $dN_{He}/dt$ ,  $dW_{heat}/dt$  [Eqs. (2), (3), and (74), respectively].
- I. Return

## III. TDGET

- A. Use TDINTP to evaluate  $\kappa$ ,  $\delta$ ,  $Z_{eff}$ ,  $\langle n_e \rangle$ ,  $n_{He}/n_e$ ,  $B_T$ ,  $I_p$ ,  $\tau_{th}$ ,  $c_{\tau_j}$  at present time-step  $t$ .
- B. If `aqcyl(i)` specified, use  $q_{cyl}(t)$  to set  $a$ ,  $R$  [Eq. (8)],
- C. Else, use TDINTP to evaluate  $a$ ,  $R$ .
- D. Call VOLARE
- E. If `cash`  $\neq 0$ , set  $\frac{n_{He}}{n_e} = \frac{N_{He}}{V\langle n_e \rangle}$ .
- F. Return

## IV. TDP AUX

- A. If first call, rescale  $P_{aux}$  waveform, determine  $B_{T,max}$ .

- B. If time dependence of  $P_{aux}$  specified, use TDINTP to evaluate  $P_{aux}$  at present time-step  $t$ ,
- C. Else, evaluate  $P_{aux} [B_T(t)/B_{T,max}]$  according to default prescription [Eq. (76)].
- D. Return

## References

- [1] D. P. Stotler and N. Pomphrey, *Fusion Technol.* 17 (1990) 577.
- [2] D. P. Stotler, R. J. Goldston, and the CIT Team, *Fusion Technol.* 20 (1991) 7.
- [3] R. E. Waltz, R. R. Dominguez, and F. W. Perkins, *Nucl. Fusion* 29 (1989) 351.
- [4] W. A. Houlberg, S. E. Attenberger, and L. M. Hively, *Nucl. Fusion* 22 (1982) 935.
- [5] N. A. Uckan and J. Sheffield, in: *Tokamak Startup*, ed. H. Knoepfel (Plenum Press, New York, 1986) p. 45.
- [6] Y. C. Sun, D. E. Post, G. Bateman, and D. Stotler, *Bull. Am. Phys. Soc.* 33 (1988) 1971.
- [7] O. Mitarai, A. Hirose, and H. M. Skarsgard, *Nucl. Fusion* 28 (1988) 2141.
- [8] N. A. Uckan, *Fusion Technol.* 14 (1988) 299.
- [9] G. Bateman, in: *Computer Applications in Plasma Science and Engineering*, ed. A. T. Drobot (Springer-Verlag, New York, 1991) p. 381.
- [10] R. J. Goldston et al., *Fusion Technol.* 21 (1992) 1050.
- [11] C. E. Singer, D. E. Post, D. R. Mikkelsen, M. H. Redi, A. Mckenney, A. Silverman, F. G. P. Seidl, P. H. Rutherford, R. J. Hawryluk, W. D. Langer, L. Foote, D. B. Heifetz, W. A. Houlberg, M. H. Hughes, R. V. Jensen, G. Lister, and J. Ogden, *Comput. Phys. Comm.* 49 (1988) 275.
- [12] S. C. Jardin, N. Pomphrey, and J. De Lucia, *J. Comput. Phys.* 66 (1986) 481.
- [13] D. P. Stotler, D. Post, and G. Bateman, *Fusion Technol.* 14 (1988) 1304.
- [14] L. M. Hively, *Nucl. Fusion* 17 (1977) 873.
- [15] B. A. Trubnikov, in: *Reviews of Plasma Physics* 7, ed. M. A. Leontovich (Consultants Bureau, New York, 1979) p. 345.
- [16] R. J. Goldston, *Plasma Phys. Controlled Fusion* 26 (1984) 87.

- [17] S. M. Kaye, Princeton Plasma Physics Laboratory, private communication (1989).
- [18] S. M. Kaye, C. W. Barnes, M. G. Bell, J. C. Deboo, M. Greenwald, K. Riedel, D. Sigmar, N. Uckan, and R. Waltz, *Phys. Fluids B* 2 (1990) 2926.
- [19] S. M. Kaye and R. J. Goldston, *Nucl. Fusion* 25 (1985) 65.
- [20] K. Odajima and Y. Shimomura, Japanese Atomic Energy Research Institute Report, JAERI-M 88-068 (1988).
- [21] P. H. Rebut, P. P. Lallia, and M. L. Watkins, Joint European Tokamak Report, JET-P(88)05 (1988) and corrigendum (1988); see also N. A. Uckan and ITER Physics Group, International Thermonuclear Experimental Reactor Report, ITER-TN-PH-0-5 (1990).
- [22] M. H. Redi, W. M. Tang, P. C. Efthimion, D. R. Mikkelsen, and G. L. Schmidt, *Nucl. Fusion* 27 (1987) 2001.
- [23] K. B. Axon, W. H. M. Clark, J. G. Cordey, M. Cox, S. J. Fielding, et al., in: *Plasma Physics and Controlled Nuclear Fusion Research, Proc. 8th Int. Conf. Brussels, 1980* (International Atomic Energy Agency, Vienna, 1981) Vol. 1, p. 413.
- [24] M. Greenwald, J. Terry, S. Wolfe, S. Ejima, M. Bell, S. Kaye, and G. H. Neilson, *Nucl. Fusion* 28 (1988) 2199.
- [25] JET Team, in: *Plasma Physics and Controlled Nuclear Fusion Research, Proc. 12th Int. Conf. Nice, 1988* (International Atomic Energy Agency, Vienna, 1989) Vol. 1, p. 41.
- [26] P-H. Rebut for the JET Team, in: *Plasma Physics and Controlled Nuclear Fusion Research, Proc. 13th Int. Conf. Washington D. C., 1990* (International Atomic Energy Agency, Vienna, 1991) Vol. 1, p. 27.
- [27] M. Keilhacker and the JET Team, *Phys. Fluids B* 2 (1990) 1291.
- [28] K. McCormick, A. Hyatt, G. Kyriakakis, E. R. Müller, H. Niedermeyer, A. Stäbler, D. Thomas, N. Tsois, ASDEX- and NI-Teams, in: *17th European Conf. Controlled Fusion and Plasma Physics, Amsterdam, 1990* (European Physical Society, 1990) Vol. 3, p. 1439.

- [29] T. Tsunematsu et al., in: Plasma Physics and Controlled Nuclear Fusion Research, Proc. 13th Int. Conf. Washington D. C., 1990 (International Atomic Energy Agency, Vienna, 1991) Vol. 3, p. 315.
- [30] D. E. Post, in: Applied Atomic Collision Physics, Vol. 2, eds. H. S. W. Massey, E. W. McDaniel, and B. Bederson (Academic Press, New York, 1984) p. 381.
- [31] D. L. Jassby, Nucl. Fusion 15 (1975) 453.
- [32] A. Sykes, M. F. Turner, and S. Patel, in: 11th European Conf. Controlled Fusion and Plasma Physics, Aachen, 1983 (European Physical Society, 1983) Vol. 2, p. 363.
- [33] F. Troyon, T. Gruber, H. Sauremann, S. Semenzato, S. Succi, Plasma Phys. Controlled Fusion 26 (1984) 209.
- [34] N. Pomphrey, Princeton Plasma Physics Laboratory, private communication (1990).
- [35] D. P. Stotler and G. Bateman, Fusion Technol. 15 (1989) 12.
- [36] A. C. Hindmarsh, in: Scientific Computing, ed. R. S. Steplman (North-Holland, Amsterdam, 1983) p. 55.
- [37] P. Colestock, Princeton Plasma Physics Laboratory, private communication (1990).
- [38] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes (Cambridge Press, New York, 1988).
- [39] P. N. Yushmanov, T. Takizuka, K. S. Riedel, O. J. W. F. Kardaun, J. G. Cordey, S. M. Kaye, and D. E. Post, Nucl. Fusion 30 (1990) 1999.
- [40] R. Snider, Bull. Am. Phys. Soc. 34 (1989) 2116; see also R. T. Snider, Nucl. Fusion 30 (1990) 2400.

## Figures

Fig. 1. Sample input file for a BPX reference discharge[10].

Fig. 2. Sample output from the steady-state calculation.

Fig. 3. Sample output from the time-dependent calculation.

Fig. 4. Plot of the time evolution of the various terms in Eq. (1) produced using data from the sample run, a BPX reference discharge[10].

```

$input
calpha = 1.0      ! Overall multiplier on P_alpha
cbrem = 1.0      !      "      "      " P_brem
ccon = 1.0       !      "      "      " P_con
coh = 1.0        !      "      "      " P_OH
csync = 0.0      !      "      "      " P_sync
cip = 0.0        ! Multiplier on I_p(q_95) expression

rminor = 0.795   ! Plasma minor radius (m)
rmajor = 2.59    !      "      major      " (m)
elong = 2.000    !      "      elongation
xtring = 0.350   !      "      triangularity
btor = 8.10      ! Toroidal magnetic field (T)
curma = 1.06e1   ! Plasma current (MA)

xpalmx = 6.39e7  ! Maximum heating power (W)
nmurak = 4       ! Maximum density is denmax
npwr = 4         ! Use xpalmx to limit P_alpha + P_aux + P_OH
ncig = 0         ! Calculate ignition margin
noptm = 1        ! Optimize choice of n and T
denmax = 5.e20   ! Maximum <n_e> (m**-3)

ltaue(1) = 1 8   ! Neo-Alcator & specified exponent tau_E scalings
cctaue(1) = 1. 1.85 ! Confinement multipliers
! ITER-P '89 scaling exponents:
txp(1) = 0.0381  ! constant
txp(2) = 0.5     ! aibar
txp(3) = 0.5     ! elong
txp(4) = 0.85    ! curent (MA)
txp(5) = 0.1     ! deneln (1.e19 m**-3)
txp(6) = 0.2     ! btor (T)
txp(7) = 0.3     ! rminor (m)
txp(8) = 1.2     ! rmajor (m)
txp(9) = -0.5    ! Pinput (MW)
lctaue = 1       ! tau_E is minimum of the two pieces
npswtc = 0       ! tau_E is written in terms of plasma energy

aibar = 2.5      ! Average mass of ionic species
xzimp = 6.0      ! Impurity charge
xzeff = 1.65     ! Effective ion charge
rnane = 0.025    ! ratio of helium to electron density
rndndt = 0.5     ! ratio of D to D + T density
resnc = 2.5      ! Neoclassical resistivity correction

lform = 3        ! Trapezoidal T, parabolic density profiles
nform = 101      ! Number of elements in radial profiles
alpn = 0.5       ! Density profile exponent
xtmixl = 0.29    ! r_mix / a for temperature profile
$end
$tdlist
ttbk(1) = 0.0 27.5 ! Time breakpoints for confinement parameters

ntnbk = 6        ! Number of breakpoints in tnbk
tnbk(1) = 0.0 7.5 8.0 11.0 17.5 27.5 ! Breakpoints for density
adenav(1) = 0.0 0.5 0.5 1.0 1.0 0.0 ! Relative density values
ntpbk = 6        ! No. of breakpoints in tpbk
tpbk(1) = 0.0 7.95 8.05 22.45 22.55 27.5 ! Breakpoints for P_aux

```

```

apaux(1) = 0.0 0.0 1.0 1.0 0.0 0.0 ! Relative P_aux values
nfindp = 0 ! Calculate Wtot(t)
nfdbak = 2 ! Feedback to maintain P_aux+P_alpha+P_OH < xpinmx
tfinal = 20.0 ! Time to end integration (s)
xpaumx = 20.0 ! Peak auxiliary power (MW)
xpinmx = 100.00 ! Feedback level (MW)
dtout = 0.5 ! Interval between output (s)
$end

```

FIG. 1

```
current = 1.0600E+07 A
density = 1.9877E+20 m**3, temperature = 9.6429E+00 keV
P-alpha = 4.1945E+07 W, P-brem = 7.7367E+06 W, P-con = 5.6163E+07 W
P-sync = 0.0000E+00 W, P-aux = 1.9955E+07 W, P-oh = 2.0001E+06 W
beta = 2.3303E-02, beta-alph = 1.2470E-03, tau-E = 9.5300E-01
c-tau = 1.8500E+00, ig. marg. = 6.5641E-01, tau-ig = 1.5647E+00
ierr = 0
```

FIG. 2

t	ne-avg	Te-avg	Palpha	Paux	Poh	Pcon	Prad	Wtot	f-DT	Wheat
2.000	0.265	1.708	0.002	0.000	2.530	1.806	0.043	0.948	0.830	0.979
2.500	0.331	1.877	0.005	0.000	3.311	2.374	0.073	1.343	0.830	2.445
3.000	0.398	2.042	0.011	0.000	4.058	2.965	0.114	1.808	0.830	4.292
3.500	0.464	2.194	0.021	0.000	4.801	3.548	0.165	2.333	0.830	6.514
4.000	0.530	2.334	0.036	0.000	5.536	4.117	0.229	2.917	0.830	9.113
4.500	0.596	2.464	0.057	0.000	6.264	4.673	0.305	3.559	0.830	12.086
5.000	0.663	2.587	0.088	0.000	6.977	5.203	0.396	4.261	0.830	15.432
5.500	0.729	2.705	0.129	0.000	7.672	5.706	0.503	5.025	0.830	19.149
6.000	0.795	2.820	0.184	0.000	8.344	6.181	0.626	5.853	0.830	23.232
6.500	0.861	2.932	0.256	0.000	8.994	6.627	0.767	6.748	0.830	27.677
7.000	0.928	3.042	0.347	0.000	9.617	7.043	0.926	7.710	0.830	32.481
7.500	0.994	3.151	0.463	0.000	10.214	7.431	1.106	8.744	0.830	37.641
8.000	0.994	3.468	0.648	10.000	8.899	8.178	1.160	9.624	0.830	42.965
8.500	1.159	5.362	3.460	20.000	4.733	12.645	1.963	17.360	0.830	56.804
9.000	1.325	6.457	7.466	20.000	3.607	15.228	2.814	23.893	0.830	71.531
9.500	1.491	7.349	13.018	20.000	2.983	19.436	3.799	30.593	0.830	88.226
10.000	1.656	7.831	18.645	20.000	2.714	26.677	4.842	36.222	0.830	107.582
10.500	1.822	7.963	23.441	20.000	2.642	32.751	5.908	40.518	0.830	129.477
11.000	1.988	7.876	27.202	20.000	2.677	37.464	6.992	43.715	0.830	153.509
11.500	1.988	8.311	30.694	20.000	2.478	41.718	7.182	46.130	0.830	179.298
12.000	1.988	8.651	33.495	20.000	2.339	45.200	7.328	48.016	0.830	206.575
12.500	1.988	8.912	35.682	20.000	2.241	47.968	7.438	49.465	0.830	235.036
13.000	1.988	9.109	37.354	20.000	2.171	50.116	7.519	50.560	0.830	264.417
13.500	1.988	9.256	38.612	20.000	2.122	51.751	7.580	51.378	0.830	294.496
14.000	1.988	9.365	39.547	20.000	2.086	52.976	7.624	51.983	0.830	325.099
14.500	1.988	9.445	40.235	20.000	2.061	53.884	7.657	52.426	0.830	356.090
15.000	1.988	9.503	40.738	20.000	2.043	54.550	7.681	52.750	0.830	387.365
15.500	1.988	9.546	41.103	20.000	2.030	55.037	7.698	52.984	0.830	418.848
16.000	1.988	9.576	41.368	20.000	2.020	55.391	7.710	53.154	0.830	450.481
16.500	1.988	9.598	41.559	20.000	2.013	55.646	7.719	53.277	0.830	482.224
17.000	1.988	9.614	41.697	20.000	2.009	55.831	7.725	53.365	0.830	514.045
17.500	1.988	9.626	41.796	20.000	2.005	55.964	7.730	53.429	0.830	545.924
18.000	1.888	11.564	45.900	20.000	1.631	64.242	6.609	52.705	0.830	578.918
18.500	1.789	13.548	45.462	20.000	1.376	69.063	5.481	49.943	0.830	612.702
19.000	1.690	15.418	41.018	20.000	1.211	68.840	4.391	45.189	0.830	645.095
19.500	1.590	17.188	34.513	20.000	1.099	64.549	3.402	39.275	0.830	674.599
20.000	1.491	18.803	27.294	20.000	1.026	62.642	2.542	32.743	0.830	700.635

Max. Paux = 20.000 MW, Final Wtot = 32.743 MJ

FIG. 3

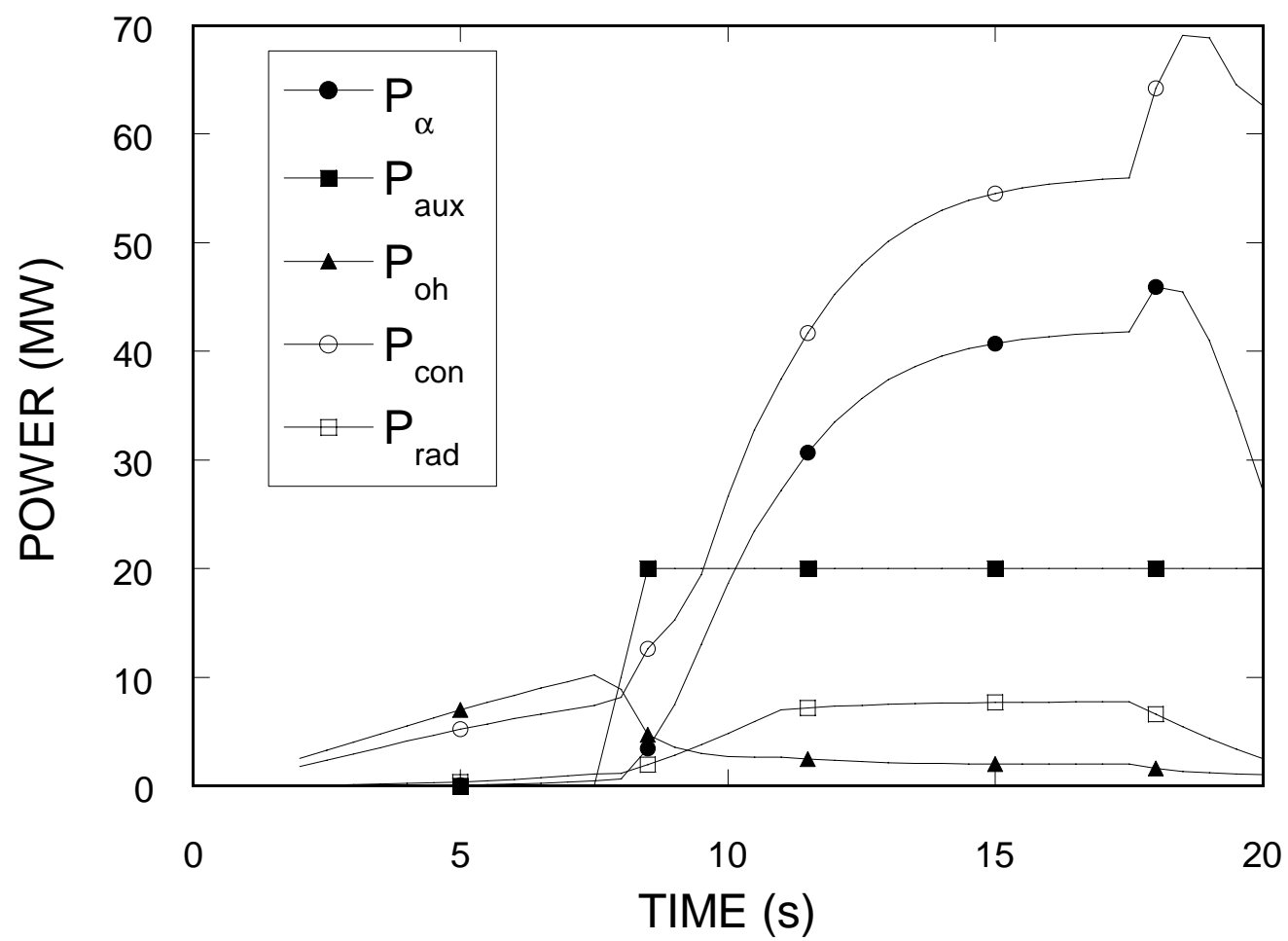


FIG. 4

Name	Default (units)	Symbol	Eq. or Sec. #	Description
aibar	2.5 (amu)	$\bar{A}_i$	Sec. 2.4	average mass of ionic species
alp <sub>j</sub>	1.5	$\alpha_j$	Eq. (12)	current density profile exponent
alpn	0.5	$\alpha_n$	Eq. (12)	density profile exponent
alpt	1.0	$\alpha_T$	Eq. (12)	temperature profile exponent
btor	8.1 (T)	$B_T$	Sec. 2.1	toroidal magnetic field
calpha	1.0	$C_\alpha$	Eq. (23)	overall multiplier on $P_\alpha$
cbrem	1.0	$C_{brem}$	Eq. (27)	overall multiplier on $P_{brem}$
ccon	1.0	$C_{con}$	Eq. (29)	overall multiplier on $P_{con}$
cctaue(j)	1.0	$c_{\tau_j}$	Eq. (32)	$j$ th ( $j = 1, 2$ ) confinement multiplier
cip	1.0	$C_{I_p}$	Eq. (11)	overall multiplier on $I_p(q_{95})$ expression
coh	1.0	$C_{OH}$	Eq. (24)	overall multiplier on $P_{OH}$
csync	0.0	$C_{sync}$	Eq. (28)	overall multiplier on $P_{sync}$
curma	10.6 (MA)	$I_p$	Sec. 2.1	plasma current
denmax	$5 \times 10^{20} \text{ (m}^{-3}\text{)}$		Sec. 3.1	maximum $\langle n_e \rangle$
elong	2.0	$\kappa$	Eq. (5)	plasma elongation
gbeta	3.0	$g_\beta$	Eq. (71)	Troyon coefficient
hugill	1.5	$h$	Eq. (57)	multiplier on Murakami form of $\bar{n}_{e,max}$
lctaue	0		Sec. 2.4	determines means of combining confinement scalings,

Name	Default (units)	Symbol	Eq. or Sec. #	Description
			Eq. (32) Eq. (33)	= 0 $\Rightarrow$ use exponent <code>xptau</code> ; otherwise $\Rightarrow$ take minimum confinement scaling
<code>lform</code>	0		Sec. 2.2	controls specification of profile form factors, = 1 $\Rightarrow$ parabolic profiles; = 2 $\Rightarrow$ trapezoidal profiles; = 3 $\Rightarrow$ trapezoidal temperature profile, parabolic density; otherwise $\Rightarrow$ parabolic profiles; > 0 $\Rightarrow$ , averages are computed numerically
<code>ltaue(j)</code>	0	$\ell_{\tau_j}$	Sec. 2.4 Eq. (34) Eq. (37) Eq. (36) Eq. (35) Eq. (40) Eq. (41)  Eq. (42) Eq. (38) Eq. (43) Eq. (45)	determines $j$ th ( $j = 1, 2$ ) $\tau_E$ scaling expression, = 1 $\Rightarrow$ neo-Alcator; = 2 $\Rightarrow$ neo-Kaye; = 3 $\Rightarrow$ Kaye - Goldston; = 4 $\Rightarrow$ Goldston; = 5 $\Rightarrow$ Kaye-All-Complex; = 6 $\Rightarrow$ Kaye-Big-Complex; = 7 $\Rightarrow$ constant = <code>tauth</code> ; = 8 $\Rightarrow$ specified exponents; = 9 $\Rightarrow$ T-10; = 10 $\Rightarrow$ Odajima-Shimomura; = 11 $\Rightarrow$ Rebut-Lallia;

Name	Default (units)	Symbol	Eq. or Sec. #	Description
				otherwise $\Rightarrow$ constant = <code>tauth</code>
<code>ncig</code>	1		Sec. 3.1	determines nature of performance calculation, = 0 $\Rightarrow$ calculates $M_I$ and $\tau_{ig}$ ; otherwise $\Rightarrow$ solves for $c_\tau$
<code>nform</code>	101		Sec. 2.2	number of elements in radial profile (form factor) arrays
<code>nmurak</code>	2		Sec. 3.1	determines formula for limiting density, = 1 $\Rightarrow$ Hugill limit; = 2 $\Rightarrow$ Greenwald limit; = 3 $\Rightarrow$ JET limit; = 4 $\Rightarrow$ specify using <code>denmax</code> ; otherwise $\Rightarrow$ calculates $\langle n_e \rangle$ using <code>npwr</code>
<code>noptm</code>	0		Sec. 3.1	optimization switch, = 0 $\Rightarrow$ carry out to calculate performance at intersection of limits; otherwise $\Rightarrow$ search along limiting boundary to find $\langle n_e \rangle$ and $\langle T \rangle_{dw}$ yielding maximum performance
<code>npswtc</code>	1		Sec. 2.4	= 0 $\Rightarrow$ $\tau_E$ is expressed as a function of $W_{tot}$ ; otherwise $\Rightarrow$ $\tau_E$ is expressed as a function of $P_{in}$
<code>npwr</code>	1		Sec. 3.1	specifies form of limiting power,

Name	Default (units)	Symbol	Eq. or Sec. #	Description
				$= 1 \Rightarrow$ limits $\langle n_e \rangle$ or $\langle T \rangle_{dw}$ so that $P_\alpha \leq \text{xpalmx}$ ; $= 2 \Rightarrow$ specifies maximum temperature via tkvmax (only if $1 \leq \text{nmurak} \leq 4$ ); $= 3 \Rightarrow$ limits $\langle n_e \rangle$ or $\langle T \rangle_{dw}$ so that $P_\alpha + P_{aux} \leq \text{xpalmx}$ ; $= 4 \Rightarrow$ limits $\langle n_e \rangle$ or $\langle T \rangle_{dw}$ so that $P_\alpha + P_{aux} + P_{OH} \leq \text{xpalmx}$ ; $= 5 \Rightarrow$ limits $\langle n_e \rangle$ or $\langle T \rangle_{dw}$ so that $P_\alpha + P_{aux} + P_{OH} - P_{rad} \leq \text{xpalmx}$ ; otherwise $\Rightarrow$ limits $\langle n_e \rangle$ or $\langle T \rangle_{dw}$ by $\beta_{max}$
nsctor	999		Sec. 2.1	number of sectors used in calculation of $A$ and $V$
q95	3.2	$q_{95}$	Sec. 2.1	specified MHD safety factor at 95% flux surface
refl	0.9	$\mathcal{R}$	Eq. (28)	synchrotron radiation reflectivity of wall
resnc	2.5	$\gamma_{NC}$	Eq. (24)	neoclassical resistivity correction
rmajor	2.59 (m)	$R$	Eq. (4)	plasma major radius
rminor	0.795 (m)	$a$	Eq. (4)	plasma minor radius
rnane	0.0	$n_{\text{He}}/n_e$	Sec. 2.5	ratio of helium to electron density
rndndt	0.5	$\frac{n_D}{n_D + n_T}$	Sec. 2.5	ratio of deuterium to deuterium plus tritium density
tauth	0.5 (s)	$\tau_{th}$	Sec. 2.4	constant confinement time for use with <code>ltaue(i) = 7</code>

Name	Default (units)	Symbol	Eq. or Sec. #	Description
tkvmax	10.0 (keV)		Sec. 3.1	maximum $\langle T \rangle_{dw}$
txp(j)	—	$t_j$	Eq. (42)	externally specified exponents for use with $l\tau_{aue}(i) = 8$ , overall constant; exponent on $\bar{A}_i$ ; exponent on $\kappa$ ; exponent on $I_p/10^6$ ; exponent on $\bar{n}_e/10^{19}$ ; exponent on $B_T$ ; exponent on $a$ ; exponent on $R$ ; exponent on $P_{in}/10^6$
txp(1)	0.0			
txp(2)	0.0			
txp(3)	0.0			
txp(4)	0.0			
txp(5)	0.0			
txp(6)	0.0			
txp(7)	0.0			
txp(8)	0.0			
txp(9)	0.0			
xdmix1	0.0		Eq. (13)	determines $r_{\text{mix}}/a$ for the density profile, $\geq 0 \Rightarrow r_{\text{mix}}/a = \text{xdmix1}$ ; otherwise $\Rightarrow r_{\text{mix}}/a = 0.27 + 1/q_{95}$ (Ref. 40)
xmi	1.0	$M_I$	Eq. (72)	desired value of $M_I$ in solving for $c_\tau$
xpalmx	$1 \times 10^8$ W		Sec. 3.1	maximum alpha or heating power
xptaue	-2.0	$x_\tau$	Eq. (32)	exponent for combining confinement scalings
xtmix1	0.0		Eq. (13)	determines $r_{\text{mix}}/a$ for the temperature profile, $\geq 0 \Rightarrow r_{\text{mix}}/a = \text{xdmix1}$ ; otherwise $\Rightarrow r_{\text{mix}}/a = 0.27 + 1/q_{95}$ (Ref. 40)

Name	Default (units)	Symbol	Eq. or Sec. #	Description
xtring	0.35	$\delta$	Eq. (4)	plasma triangularity
xzeff	1.65	$Z_{eff}$	Eq. (53)	effective ion charge
xzimp	6.0	$Z$	Sec. 2.5	impurity charge
ydedgl	0.0		Eq. (13)	density pedestal at edge $n_{edge}/n_0$
ytedgl	0.0		Eq. (13)	temperature pedestal at edge $T_{edge}/T_0$

Name	Default (units)	Symbol	Eq. or Sec. #	Description
abtor(i) abtor(1) abtor(2) abtor(3) abtor(4) abtor(5)	— (T) btor * zztf2 btor * zztf4 btor btor btor * zztf3	$B_T(t)$	Sec. 3.2	plasma current at time t <b>ibk</b> (i), $i \leq$ nt <b>ibk</b>
actaue(i, j)	$c_{\tau_j, ss}$	$c_{\tau_j}(t)$	Sec. 3.2	confinement time multipliers ( $j = 1, 2$ ) at time t <b>tbk</b> (i), $i \leq$ nt <b>tbk</b>
acur <b>nt</b> (i) acur <b>nt</b> (1) acur <b>nt</b> (2) acur <b>nt</b> (3) acur <b>nt</b> (4) acur <b>nt</b> (5)	— (A) 0.0 $I_{p, ss} \frac{\mathbf{tnull}}{\mathbf{tramp}}$ $I_{p, ss}$ $I_{p, ss}$ 0.0	$I_p(t)$	Sec. 3.2	plasma current at time t <b>ibk</b> (i), $i \leq$ nt <b>ibk</b>
adenav(i)  adenav(1) adenav(2) adenav(3)	—  0.0 1.0 1.0	$\frac{\langle n_e \rangle}{\langle n_e \rangle_{ft}}(t)$	Sec. 3.2	relative (to den <b>flt</b> ) average electron density at time t <b>nbk</b> (i), $i \leq$ nt <b>nbk</b>

Name	Default (units)	Symbol	Eq. or Sec. #	Description
adnav(4)	0.0			
aelong(i)	—	$\kappa(t)$	Sec. 3.2	elongation at time $\text{tbk}(i), i \leq \text{ntbk}$
aelong(1)	1.0			
aelong(2)	$0.7 * \text{elong}$			
aelong(3)	elong			
aelong(4)	elong			
aelong(5)	1.0			
apaux(i)	0.0	$\frac{P_{aux}}{P_{aux,max}}(t)$	Sec. 3.2	relative (to xpaumx) auxiliary power at time $\text{tpbk}(i), i \leq \text{ntpbk}$
aqcyl(i)	0.0		Sec. 3.2	cylindrical safety factor at time $\text{tbk}(i), i \leq \text{ntbk}$
armaj(i)	$\text{rmajor} + \text{rminor} - \text{armin}(i)$	$R(t)$	Sec. 3.2	major radius at time $\text{tbk}(i), i \leq \text{ntbk}$
armin(i)	— (m)	$a(t)$	Sec. 3.2	minor radius at time $\text{tbk}(i), i \leq \text{ntbk}$
armin(1)	0.0			
armin(2)	rminor			
armin(3)	rminor			
armin(4)	rminor			
armin(5)	0.0			
arnane(i)	rnane	$\frac{n_{He}}{n_e}(t)$	Sec. 3.2	helium ash concentration at time $\text{tnbk}(i), i \leq \text{ntnbk}$
atauth(i)	tauth (s)	$\tau_{th}(t)$	Sec. 3.2	specified energy confinement time at time $\text{ttbk}(i), i \leq \text{nttbk}$

Name	Default (units)	Symbol	Eq. or Sec. #	Description
atring(i) atring(1) atring(2) atring(i) atring(i) atring(i)	— 0.0 $0.2 * \text{xtring}$ xtring xtring 0.0	$\delta(t)$	Sec. 3.2	triangularity at time $\text{tbk}(i)$ , $i \leq \text{ntbk}$
azeff(i)	xzeff	$Z_{eff}(t)$	Sec. 3.2	effective ion charge at time $\text{tnbk}(i)$ , $i \leq \text{ntnbk}$
cash	0.0	$C_{ash}$	Eq. (3)	multiplier on source and sink in $dN_{He}/dt$ equation
denflt	$\langle n_e \rangle_{ss} (\text{m}^{-3})$	$\langle n_e \rangle_{ft}$	Sec. 3.2	maximum volume-averaged density
dtout	0.2 (s)		Sec. 3.2	length of output time-steps
nfdbak	— $\text{npwr} - 2$ 0		Sec. 3.2	determines feedback mode; if $3 \leq \text{npwr} \leq 5$ , otherwise; $= 1 \Rightarrow P_{aux} + P_\alpha \leq P_{in,max}$ ; $= 2 \Rightarrow P_{aux} + P_\alpha + P_{OH} \leq P_{in,max}$ ; $= 3 \Rightarrow P_{aux} + P_\alpha + P_{OH} - P_{rad} \leq P_{in,max}$ ; otherwise $\Rightarrow$ no feedback
nfindp	0		Sec. 3.2	determines nature of time-dependent calculation, $= 0 \Rightarrow$ calculates $W_{tot}(t)$ ; otherwise $\Rightarrow$ solve for $\text{xpaumx}$ required to reach $\text{wtotf}$

Name	Default (units)	Symbol	Eq. or Sec. #	Description
ntbk	5		Sec. 3.2	number of breakpoints in t <b>bk</b>
ntibk	5		Sec. 3.2	number of breakpoints in t <b>ibk</b>
ntnbk	4		Sec. 3.2	number of breakpoints in t <b>nbk</b>
ntpbk	0		Sec. 3.2	number of breakpoints in t <b>pbk</b>
nttbk	2		Sec. 3.2	number of breakpoints in t <b>tbk</b>
tauphe	$10^6$ (s)	$\tau_{p,He}$	Eq. (3)	helium ash particle confinement time
t <b>bk</b> ( <i>i</i> )	— (s)		Sec. 3.2	time breakpoint array for plasma shape, $i \leq \text{ntbk}$
t <b>bk</b> (1)	0.0			
t <b>bk</b> (2)	1.5			
t <b>bk</b> (3)	tramp			
t <b>bk</b> (4)	t <b>bk</b> (3) + t <b>burn</b>			
t <b>bk</b> (5)	t <b>bk</b> (4) + t <b>dump</b>			
t <b>final</b>	tramp + t <b>burn</b> (s)	$t_f$	Sec. 3.2	time at which integration is to end
t <b>ibk</b> ( <i>i</i> )	— (s)		Sec. 3.2	time breakpoint array for fields, $i \leq \text{ntibk}$
t <b>ibk</b> (1)	0.0			
t <b>ibk</b> (2)	t <b>null</b>			
t <b>ibk</b> (3)	tramp			
t <b>ibk</b> (4)	t <b>ibk</b> (3) + t <b>burn</b>			
t <b>ibk</b> (5)	t <b>ibk</b> (4) + t <b>dump</b>			
t <b>init</b>	1.5 (s)	$t_{init}$	Sec. 3.2	time at which integration is to begin
t <b>kvini</b>	2.0 (keV)		Sec. 3.2	initial temperature of plasma

Name	Default (units)	Symbol	Eq. or Sec. #	Description
tnbk(i)	— (s)		Sec. 3.2	time breakpoint array for plasma density, $i \leq \text{ntnbk}$
tnbk(1)	0.0			
tnbk(2)	tnbk(1) + tramp			
tnbk(3)	tnbk(2) + tburn			
tnbk(4)	tnbk(3) + tdump			
tpbk(i)	— (s)		Sec. 3.2	time breakpoint array for auxiliary power, $i \leq \text{ntpbk}$
ttbk(i)	— (s)		Sec. 3.2	time breakpoint array for confinement parameters, $i \leq \text{nttbk}$
ttbk(1)	0.0			
ttbk(2)	tramp + tburn + tdump			
wtotf	$W_{tot,ss}$ (J)	$W_{tot,f}$	Sec. 3.2	plasma energy to be reached at tfinal
xnhein	0.0	$N_{\text{He},0}$	Eq. (3)	initial number of helium ash particles
xpaumx	10.0 (MW)	$P_{aux,max}$	Sec. 3.2	maximum value to be attained by $P_{aux}$
xpinmx	— (MW) $\frac{\text{xpalmx}}{10^6} + 0.3$ 0	$P_{in,max}$	Sec. 3.2	maximum input power allowed during feedback; if $3 \leq \text{npwr} \leq 5$ , otherwise